COMPUTER BLAST DESIGN FOR LARGE DIAMETER BLASTHOLE STOPES UNDERGROUND
- FINAL REPORT -

PREPARED FOR

DEPARTMENT OF
SUPPLY AND SERVICES CANADA
1SQ83 - 00153
3-9186

BY

A. BAUER
W. A. CROSBY
T. KONIARIS
P. KATSABANIS
L. MACDONALD

MINING RESOURCE ENGINEERING LIMITED
8 PRINCESS STREET,
KINGSTON, ONTARIO,
CANADA - K7L 1A2

175 pp

JANUARY, 1985

MINING RESOURCE ENGINEERING LIMITED

CONSULTING ENGINEERS

8 Princess Street, Kingston, Ontario
Canada K7L 1A2 Telephone 613-546-4089

COMPUTER BLAST DESIGN FOR LARGE DIAMETER BLASTHOLE STOPES UNDERGROUND

FINAL REPORT

MINING RESOURCE ENGINEERING LIMITED
8 PRINCESS STREET
KINGSTON, ONTARIO
CANADA
K7L 1A2

OCT 23 1985
MICROMEDIA

JANUARY 31, 1985

TABLE OF CONTENTS

		PAGE
	LIST OF FIGURES	(iii)
1.0	SUMMARY	1
2.0	INTRODUCTION	2
2.1	Blast Design Approach	6
3.0	INSTRUMENTATION	12
3.1	Background to the Instrumentation Requirement	12
3.2	Benefits of the Instrumentation	13
3.3	Overview of Selected Geophysical Techniques	14
4.0	COMPUTER DESIGN THEORY AND COMPUTER EQUIPMENT	16
4.1	Theory Used	16
4.2	Vibration Checks	18
4.3	Comparison Between Usually Accepted Design Practice and Reality	20
4.4	Blast Design Flowchart	22
4.5	Computer Software and Hardware	30
5.0	USERS MANUAL	31
5.1	Input Data	31
5.2	Data for the VCRM Program	32
5.3	Data for Program LDBB	33
5.4	Data for Graphics Programs	35
6.0	TYPICAL PROGRAM OUTPUT	36
7.0	CONCLUSION	48
	BIBLIOGRAPHY	49

TABLE OF CONTENTS (continued)

PAGE

APPENDIX A..... A1

APPENDIX B..... A116

APPENDIX C..... A137

LIST OF FIGURES

FIGURE		PAGE
1	COMPOSITE PLAN VIEW OF BLASTHOLE LAYOUT.....	4
2	PLAN VIEW OF THE LOCATIONS OF THE BOTTOM OF THE BLASTHOLES FROM FIGURE 1.....	5
3	THREE DIMENSIONAL VIEW OF A TYPICAL BLAST.....	8
4	THREE DIMENSIONAL VIEW OF A TYPICAL BLAST AS PRESENTLY DELAYED SHOWING EACH HOLE COMPLETELY FIRED PRIOR TO SUBSEQUENT HOLE FIRINGS.....	9
5	THREE DIMENSIONAL VIEW OF A BLAST SHOWING MORE THAN ONE HOLE FIRING AT ANY ONE TIME.....	11
6	TYPICAL SCALED CRATER DEPTH - DEPTH RATIO CURVE.....	19
7	COMPARISON BETWEEN THE GENERALLY ACCEPTED DESIGN PRACTICE AND TWO ACTUAL CASES.....	21
8	MAXIMUM DISTANCE FROM A CHARGE IN WHICH ROCK IS BROKEN.....	23
9(A)	THE FIRST PART OF THE LARGE DIAMETER BLAST-HOLE BENCH BLAST DESIGN PROGRAM FLOWCHART.....	24
9(B)	THE SECOND PART OF THE LARGE DIAMETER BLASTHOLE BENCH BLAST DESIGN PROGRAM FLOW-CHART.....	25
10	THE VERTICAL CRATER RETREAT METHOD BLAST DESIGN PROGRAM FLOWCHART.....	29
11	TYPICAL COMPUTER OUTPUT SHOWING BLASTHOLE PROJECTION ON PLANE XZ.....	38
12	TYPICAL COMPUTER OUTPUT SHOWING HORIZONTAL BLASTHOLE PROJECTION.....	39
13	TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING PLAN VIEW OF A STOPE.....	40
14	TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A ROW CROSS-SECTION WITH CHARGE DEPTHS.	41

LIST OF FIGURES

FIGURES		PAGE
15	TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A ROW CROSS-SECTION WITH CHARGE WEIGHTS AND DELAYS.....	42
16	TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A COLUMN SECTION WITH CHARGE DEPTHS....	43
17	TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A COLUMN SECTION WITH CHARGE WEIGHTS AND DELAYS.....	44
18	TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING STOPE PLAN.....	45
19	TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING A ROW SECTION WITH CHARGE WEIGHTS, DELAYS AND CHARGE DEPTHS.....	46
20	TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING A COLUMN SECTION WITH CHARGE WEIGHTS, DELAYS AND CHARGE DEPTHS.....	47

1.0 SUMMARY

This report presents a computer program prepared by Mining Resource Engineering Limited which is capable of designing underground blasts using large diameter blastholes. The program produces designs for either large diameter blasthole bench blasting or for the vertical crater retreat method. In particular, the program takes into account the blasthole wander that has proved a significant problem in these types of blast design.

The input to the program includes three dimensional descriptions of the blasthole locations in the blast, geological data such as cratering results, powder factor requirements, etc. for different material types, standard explosives data on strength, density, etc. and information on points in the mine requiring protection from excessive vibrations.

The output gives horizontal and/or vertical projections of the blastholes, plans and cross-sections of the blast and complete loading instructions for the blaster.

1.0 SOMMAIRE

Ce rapport présente un programme d'ordinateur capable de faire le design d'explosions souterraines utilisant des trous d'explosifs de grand diamètre. Le programme a été préparé par Mining Resource Engineering Limited. Il produit des designs d'explosion soit pour des trous d'explosifs de grand diamètre, pour des explosions par étage (bench blasting) ou pour la méthode de cratère refoulé verticalement (vertical crater retreat). En particulier le programme tient compte de la divagation des trous d'explosifs, ce qui s'est avéré un problème important dans le design de ces types d'explosions.

L'information requise par le programme comprend: une description tridimentionnelle de la position des trous d'explosifs, l'information géologique telle que les résultats sur la formation des cratères, la valeur du facteur poudre, etc. pour les différents types de materiau, l'information habituelle sur la force des explosifs, la densité, etc. et l'information sur les endroits dans la mine qui ne doivent pas être exposés à des vibrations excessives.

Le rapport du programme donne des projections horizontales et/ou verticales des trous d'explosifs, les plans et vues en coupe de l'explosion ainsi que les instructions de charge complètes pour le dynamiteur.

2.0 INTRODUCTION

The use of large diameter blastholes is becoming more common in underground operations to allow mines to take advantage of the economy of increased blast size and corresponding reduced operational delays and disruptions. To improve economy, high stope heights are more frequently being blasted. Following the initial development blasts, produced using slots or circular openings and/or undercutting, two main primary blasting methods are used, these being crater retreat and regular bench blasting. The crater retreat method utilizes single cratering charges near the bottom of each hole whereas the bench blast uses the entire hole for the blast usually loaded with multiple decks.

As bench heights, and thus hole depths, become larger, the degree of hole wander increases markedly. With depths of up to 30 m, hole sizes of from 125mm to 150mm exhibit some deviation, although not to such an extent that blast design is seriously affected. With increasing depth, however, hole wander becomes more significant. The result is erratic burdens and spacings being produced at the base of the blast. In some areas of the blast, where holes have converged, the blast may be greatly overloaded whereas elsewhere, where holes have diverged, burdens may be so large as to be nearly unmovable, or certainly of a size that would produce very poor fragmentation.

There is even the possibility of decks being fired out of sequence when holes converge excessively.

There are a number of instruments available which are already being used to determine the actual direction of the blastholes giving inclination and orientation with depth. These include the Gyroprobe produced by Humphries, San Diego, Ca. and the Abem Reflex Fotobar produced by Atlas Copco, Sweden. The typical accuracy of these units is of the order of 0.2 m per 100m of hole depth.

At present these measuring instruments typically have digital readout systems, but it will be a relatively simple process to provide direct computer input for the hole data, possibly by recording on magnetic tape.

Figure 1 shows a composite plan view of a blasthole layout for an actual bench blast illustrating both the locations of the blasthole collars and the blasthole toes. The hole wander is represented by the parallel dashed lines joining the top and bottom of the hole. As can be seen, there is generally no overall trend for hole direction from one hole to the next. The major contributing factors to the hole deviation are poor hole collaring, the drill and drilling techniques as well as the geological conditions.

Figure 2 highlights the result of the random hole directions. This figures presents a plan view of the bottom of the blastholes from Figure 1. The erratic spacings and burdens clearly indicate the difficulty in charge deck size determination and delay time selection.

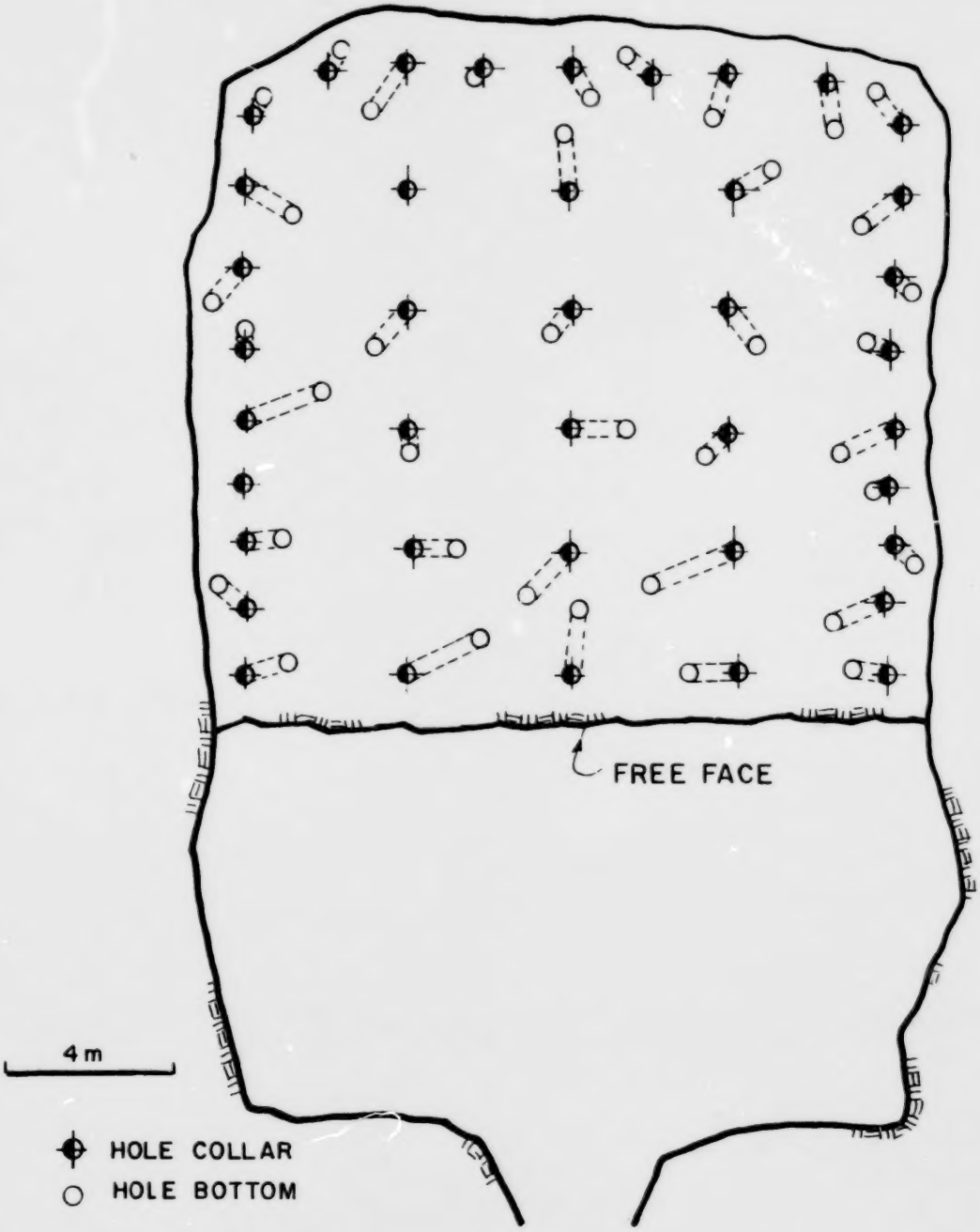


FIGURE 1 : COMPOSITE PLAN VIEW OF BLASTHOLE LAYOUT.

FIGURE 2 : PLAN VIEW OF THE LOCATIONS OF THE BOTTOM OF THE BLASTHOLES FROM FIGURE 1.

2.1 Blast Design Approach

A computer program has been written which will first produce a graphical model of the entire blast. The model is capable of including all of the relevent rock structure and geology. The program then calculates the optimum blast design for the particular hole layout and hole deviations. Computer output is both graphical, showing blast plans and sections with charge locations, and digital to present charge explosive type, size, delay period, etc.

One of the first advantages of the computer design approach is the speed at which the blast design can be prepared. The recorded blasthole data can be fed into the program during the drilling operation, the result being that in the extreme cases when extra holes are required, this is known almost immediately. The extra holes can then be drilled prior to the drill rig leaving the area, which therefore gets away from the costly business of returning a drill to the blast at a later time.

The use of the hole deviation measuring instrument is particularly useful for mines not operating with an undercut. At present these mines have virtually no knowledge of the actual explosive distribution at the base of their blasts.

The computer program will first design the development blasts for operations utilizing a benching technique. This blast can be designed to a circular opening or a rectangular raise with the charge weight being related to the standoff

distance which will also be related to the diameter or size of the opening. This is similar to the design to open up a drift round.

The computer program designs the primary blasts considering such factors as:

(1) The maximum burden that can be broken for the given material and explosive type.

(2) The maximum size of explosive deck which is dictated by the vibration experienced by pillars, etc.

(3) The rock strength, work index, explosive consumption requirements given the required fragmentation desired and the preferred direction of shooting versus rock structure, etc.

(4) Short burdens are checked for the requirement of extra decking.

(5) Wall hole design is incorporated to produce acceptable stope sides.

(6) Minimum deck separation distances are calculated.

(7) A simulation of the blast firing is programmed to select interhole delays as well as interdeck delays. Figure 3 shows a three dimensional view of a typical blast. Blasthole wander can be seen to cause quite varying burdens near the bottom of the blastholes. Figure 4 shows the typical blasting sequence presently being used where one row is fired and only one hole is detonated at any one time. In this case the bottom deck of a hole would first be initiated, followed by the second, third, fourth deck, etc. and only when the entire hole has been fired would the bottom deck on the next hole be shot. This initiation method has the problem that long delays

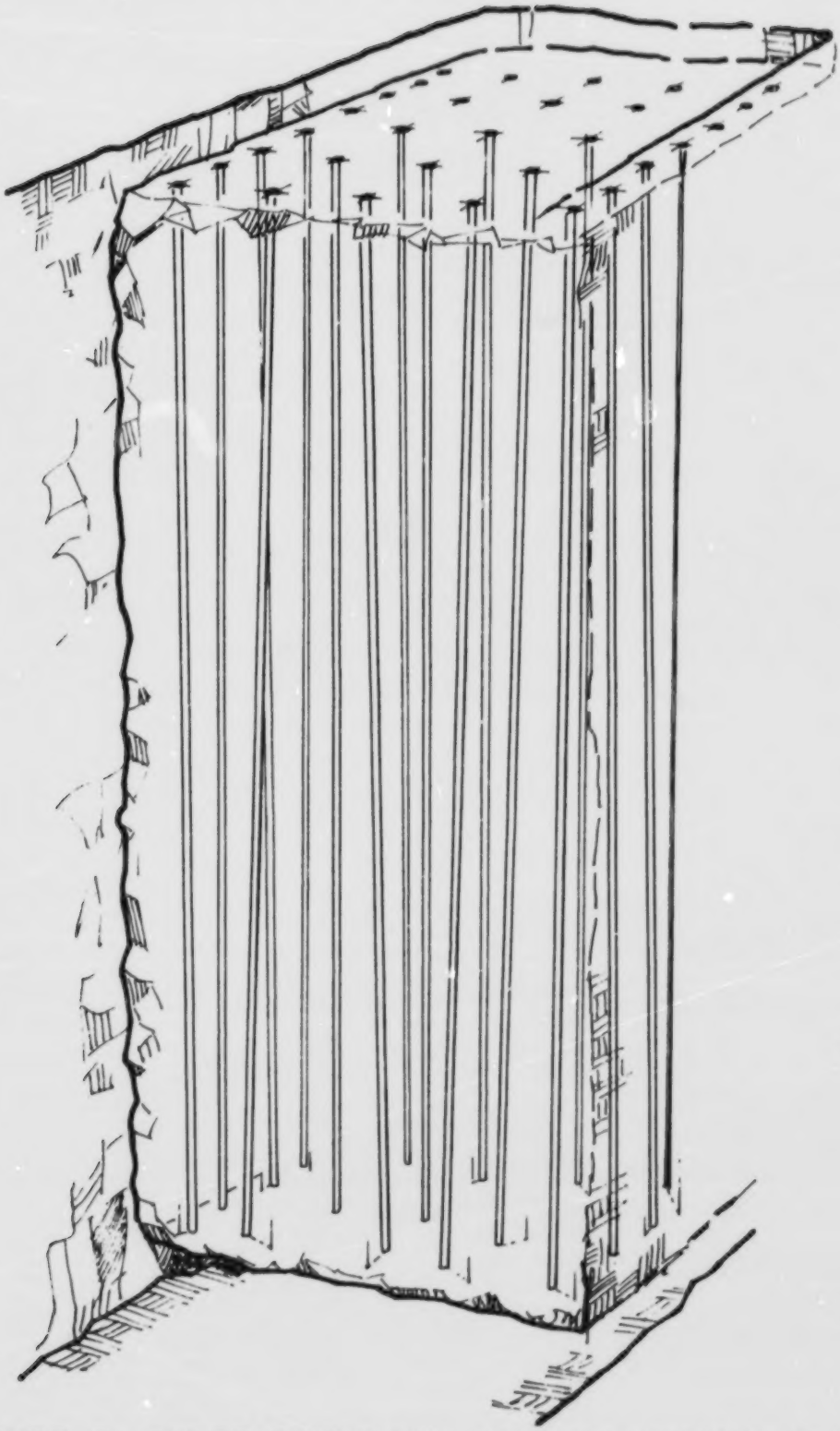


FIGURE 3: THREE DIMENSIONAL VIEW OF A TYPICAL BLAST.

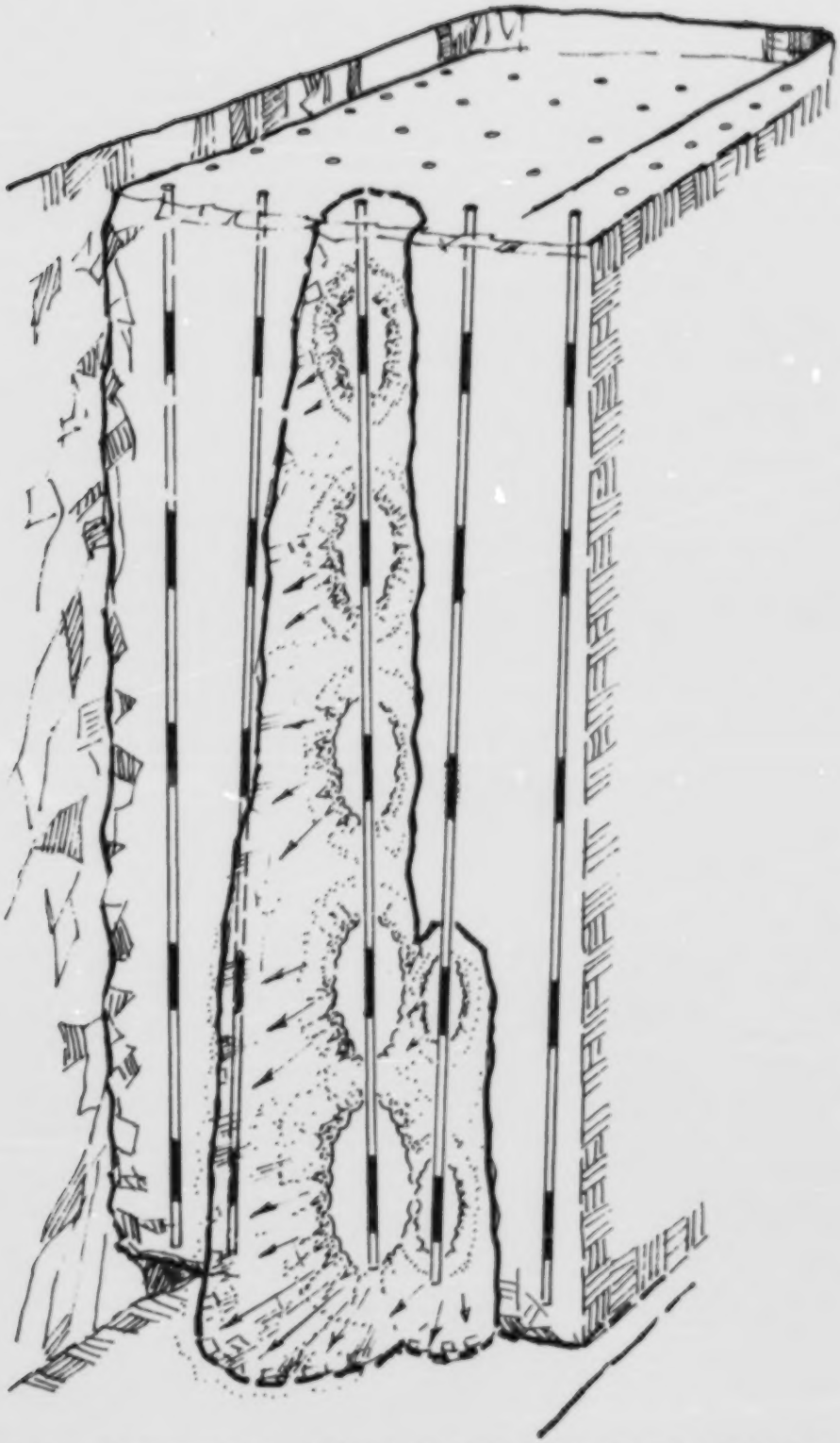


FIGURE 4 : THREE DIMENSIONAL VIEW OF A TYPICAL BLAST AS PRESENTLY DELAYED SHOWING EACH HOLE COMPLETELY FIRED PRIOR TO SUBSEQUENT HOLE FIRINGS.

may exist between adjacent explosive decks.

Figure 5 presents an alternative approach for a single row blast. After firing two decks in the initial cut hole the next three delay periods would fire the third deck in the cut hole followed by the bottom decks in each adjacent hole. This approach is used in the design program.

In open pit blasting it is frequently found that single row blasting gives poor fragmentation. It is therefore the approach that blasting schemes are developed to safely fire multiple row shots for this type of underground blasting. (8) The design is performed to prevent overbreak at both the top and bottom of the shot.

(9) For crater retreat blasting, the charge sizes and locations are designed, using cratering curves, relative to the blasthole spacings. The appropriate charge delays are also selected.

This type of computer blast design program should prove to be an invaluable tool for the blasting engineer to first quickly determine if the blast has been adequately drilled off and then to give the complete layout of deck charge size, location and delay period.

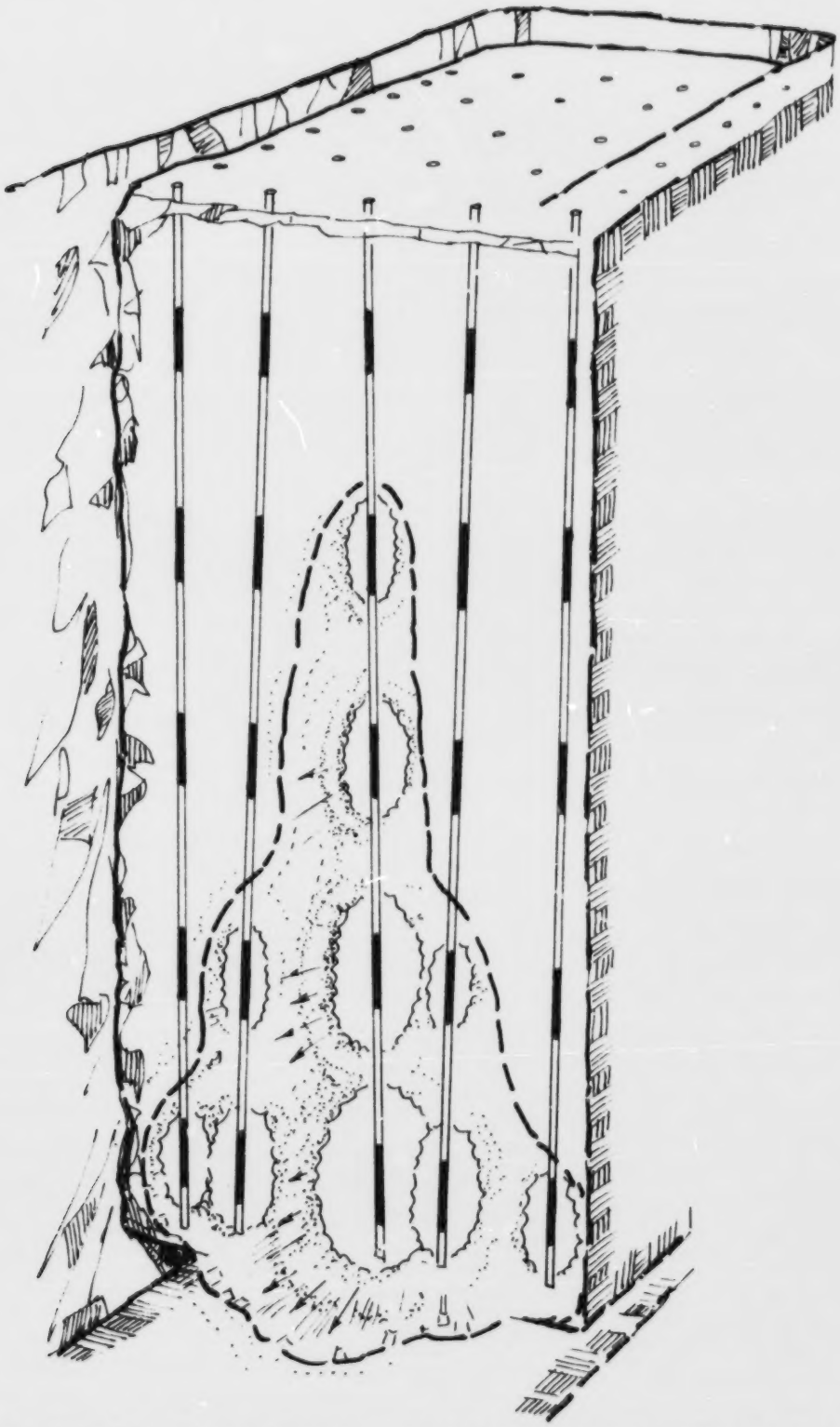


FIGURE 5: THREE DIMENSIONAL VIEW OF A BLAST SHOWING MORE THAN ONE HOLE FIRING AT ANY ONE TIME.

3.0 INSTRUMENTATION

Along with the current instrumentation presently being used in the field to measure and describe the position of the blasthole within the blast itself, the writing of this program identified the need for instrumentation that will also describe the burden on front row holes in the bench blasting operation.

The program, as presented, accepts two forms of input as regards front row burdens. These may be inputted from a specific measuring instrument, should one be developed. Alternatively a constant burden may be inputted in the form of a plane face, this being the one presented in this report.

This section describes the reasons for the requirement of a burden measuring instrument, the types of technology that may be employed in its development, and a recommendation as to the best approach for the instruments development in the opinion of the authors.

3.1 Background to the Instrumentation Requirement

This computerized approach to blast design will be considerably enhanced if all the major factors that influence the efficiency of a blast can be taken into account. One such major factor is the burden associated with the front row of drill holes immediately adjacent to the free face, as commonly only one row of blastholes may be fired at one time. Even if this is not the case, it is information that, if known, can greatly improve the blast design process. Owing to

the blasthole deviations and an uneven free face (often possessing overhangs), front-row burden values can vary significantly in a complex manner at various depths within the same hole and/or between adjacent holes. Complex variability can easily be accommodated using these computer design techniques provided that the required spatial relationships between hole location and topography of the free face are known. The acquisition of such data is, however, problematic unless a relatively rapid geophysical technique can be developed in order to suit this novel application. This section details an investigation to assess the viability of different geophysical techniques for providing data on front-row burdens in a compatible form that can be incorporated into the computer blast-design program that is now developed.

3.2 Benefits of the Instrumentation

Failure to adequately determine front-row burdens has obvious economic ramifications. Overestimations can lead to the uneconomic use of excessive explosive charge weights whereas underestimations are likely to cause poor fragmentation with a consequent need for subsequent blasting. It is clear that such potential errors serve to detract considerably from the major cost-saving advantage offered by the computerized approach to blast design.

3.3 Overview of Selected Geophysical Techniques

Four geophysical techniques were evaluated in order to assess their potential usefulness for determining front-row burdens. These techniques are:

(i) seismic reflection

(ii) radar reflection

(iii) gravity, and

(iv) acoustic reflection.

Owing to the likelihood of inaccessible mining faces (upon which targets for conventional surveying technique may not be placed), only borehole tools were considered suitable. Application of borehole seismic, radar and acoustic techniques would entail measuring the return travel time of an impulse signal which is generated in the drill hole and reflected at the rock/air interface. The gravity method, on the other hand, would involve the measurement of small differences in gravity caused by the irregular relief of the free face.

Relatively low frequency conventional seismic reflection is discounted owing to the very short travel times that are anticipated. For a distance of 4 to 8 metres between the blasthole and free face and, using an assumed average velocity of 6000 metres/sec, a two-way travel time of less than 3 millisecs is realized. Travel times of this small order cannot be resolved using current borehole seismic equipment.

Under conditions where conductive ore (of the order of 1 ohm-m resistivity) is being mined , use of the radar technique is inadvisable. Such conductive ores (ie. nickel ores) are essentially non-transparent to radar electromagnetic frequencies of an order of 100 mhz, penetration being less than approximately 0.6 metres.

Measurement precision of present borehole gravity tools is only $\pm$ 1 mgel. This order of precision is not sufficient to give an accurate picture of the 'irregularity' of the mine face. More over, localized changes in density of the ore close to the blasthole are likely to mask any variations that may be due to changes in relief of the mining face.

The acoustic technique is the most promising of the four geophysical techniques evaluated and does not suffer the limitations of the other three. Moreover, within recent years it has been used successfully in inter-borehole applications (eg. Wong et al, 1983; Aki et al, 1984, McCann et al, 1975) that are closely related to our proposed burden determination investigation.

It is therefore recommended that this be the approach taken in the development of a suitable burden measuring device to be used in conjunction with the computer blast design program as described in this report.

4.0 COMPUTER DESIGN THEORY AND COMPUTER EQUIPMENT

4.1 Theory Used

For the prediction of the craters which are formed because of the detonation of a charge, Livingston's crater theory was used, (Bauer, 1961 & Livingston, 1973). The basic points of this theory are given in the following. When an explosive charge of constant weight and shape is placed at different distances from a flat free face and detonated, the amount of material excavated changes with the depth of burial of the charge.

At deep depths the explosion is contained and all of the energy is used up in local crushing and seismic waves. As the depth of charge is reduced, surface slabbing is produced. Near the point where failure just starts, large pie-shaped fragments are formed in a massive material and the crater is quite shallow. As the depth of the charge is further reduced, the shape and depth of the crater change as does the degree of fragmentation. In addition to the tensile slabs, a series of radial cracks run out from the borehole. These are due to failures starting near the explosion and which run up through the material. There is also a series of concentric circular cracks with the borehole as center. The crater dimensions and the degree of fragmentation change continuously with the depth of burial of the charge. Associated with a certain degree of fragmentation there is also a certain level of noise and flyrock

associated with the shock. In addition as the depth of a given weight of charge is reduced, the quantity of material broken passes through a maximum and at this point the degree of fragmentation is good. It is this optimum point which is of interest for production shooting. At this point flyrock is minimized and there is little noise or airblast. That is there is a minimum of wasted energy. After obtaining the optimum point for a small weight of explosive, it is then necessary to scale up to a much larger charge weight and be able to predict the results.

The following parameters are of major importance to the theory:

(a) Depth of burial of the charge, (d_c). This is the distance from the centre of the charge to the free surface. The depth at which the surface failure just starts is called the critical depth of burial (N).

(b) Strain energy factor, (E). Livingston found that the critical depth of burial is proportional to the one-third power of the charge weight ($W^{1/3}$). The constant of proportionality (E) is called the strain energy factor. If the factor E is known the critical depth for any charge weight can be calculated.

ie. $N \propto W^{1/3}$ and $N = E\,W^{1/3}$ ..... (1)

(c) The depth ratio (Δ) is the ratio of the depth of burial to the critical depth for the same weight of charge.

ie. $\Delta = dc/N$ or $dc = \Delta N$..... (2)

Hence the general equation for charge depth of burial from equations (1) and (2) is given by:

$$dc = \Delta N = \Delta E\,W^{1/3}$$

The depth ratio can be correlated to any scaled parameter of the crater such as the scaled crater depth or the scaled crater radius. The shape of the resulting curves is shown in Figure 6. This figure gives the scaled crater depth as a function of the depth ratio. It follows that the deepest crater results at a depth ratio of approximately 0.3. At this depth ratio the fragmentation is good and the use of the energy of the explosive optimum. It follows that for optimum blasting results the charge has to be such that the depth ratio is close to the optimum depth ratio.

4.2 Vibration Checks

In recent years the question of what constitutes a valid criterion for blasting damage has been discussed by a number of authors. Extensive statistical reviews have been made of published vibration data to determine which of the blast induced parameters, such as particle displacement, velocity or acceleration,best measures the damaging effects of a vibrational wave on a structure. The current conclusion is that damage can best be correlated with the peak particle velocity of the wave regardless of the direction of motion; longitudinal, vertical or transverse.

Historically several formulae have been developed to give peak particle velocity. The formula which is used in the programs is widely accepted and was developed by Devine, (Devine, 1966). According to this formula:

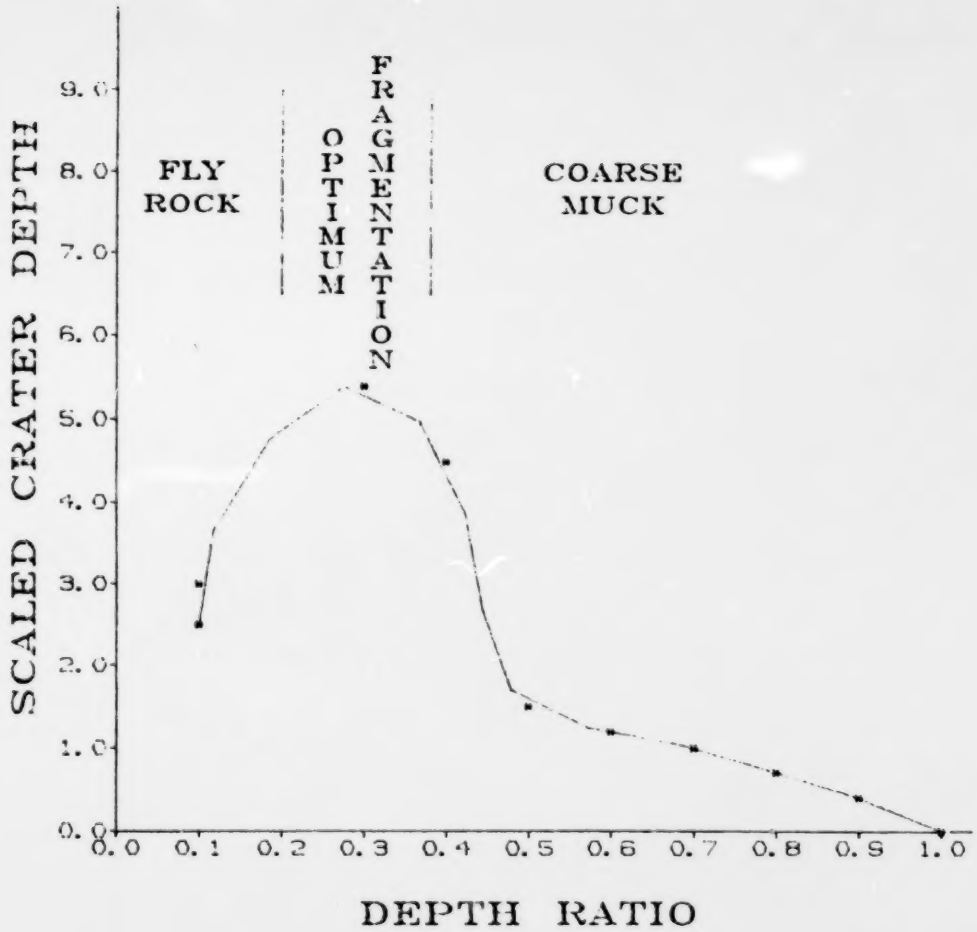


FIGURE 6 : TYPICAL SCALED CRATER DEPTH - DEPTH RATIO CURVE.

$$V = k(R/W^{1/3})^m$$

where V the peak particle velocity

R the distance between the explosion and the recording site

W the weight of the charge

k, m site factors which must be determined at the particular mine site.

4.3 Comparison Between Usually Accepted Design Practice and Reality

One of the major disadvantages of the application of Livingston's theory today is that the generally accepted design practice tends to ignore the fact that a charge detonating at an earlier period creates a free face for the subsequent detonations. The generally accepted case which is shown at the top of Figure 7 assumes that the only active free face is the initial one.

However the actual case resembles the second drawing of Figure 7. Moreover if the charge detonating at a later period is very close to the free face created by an earlier detonation, the result will be flyrock and waste of energy. Therefore charge 2 (Figure 7 third drawing) will not break the volume of rock which it was supposed to break, leaving excessive burdens for the next charge.

In the programs presented in this report an attempt was made to simulate the reality.

At each depth of burial (AB) (depth from the closest free face) the crater radius can be calculated by using the

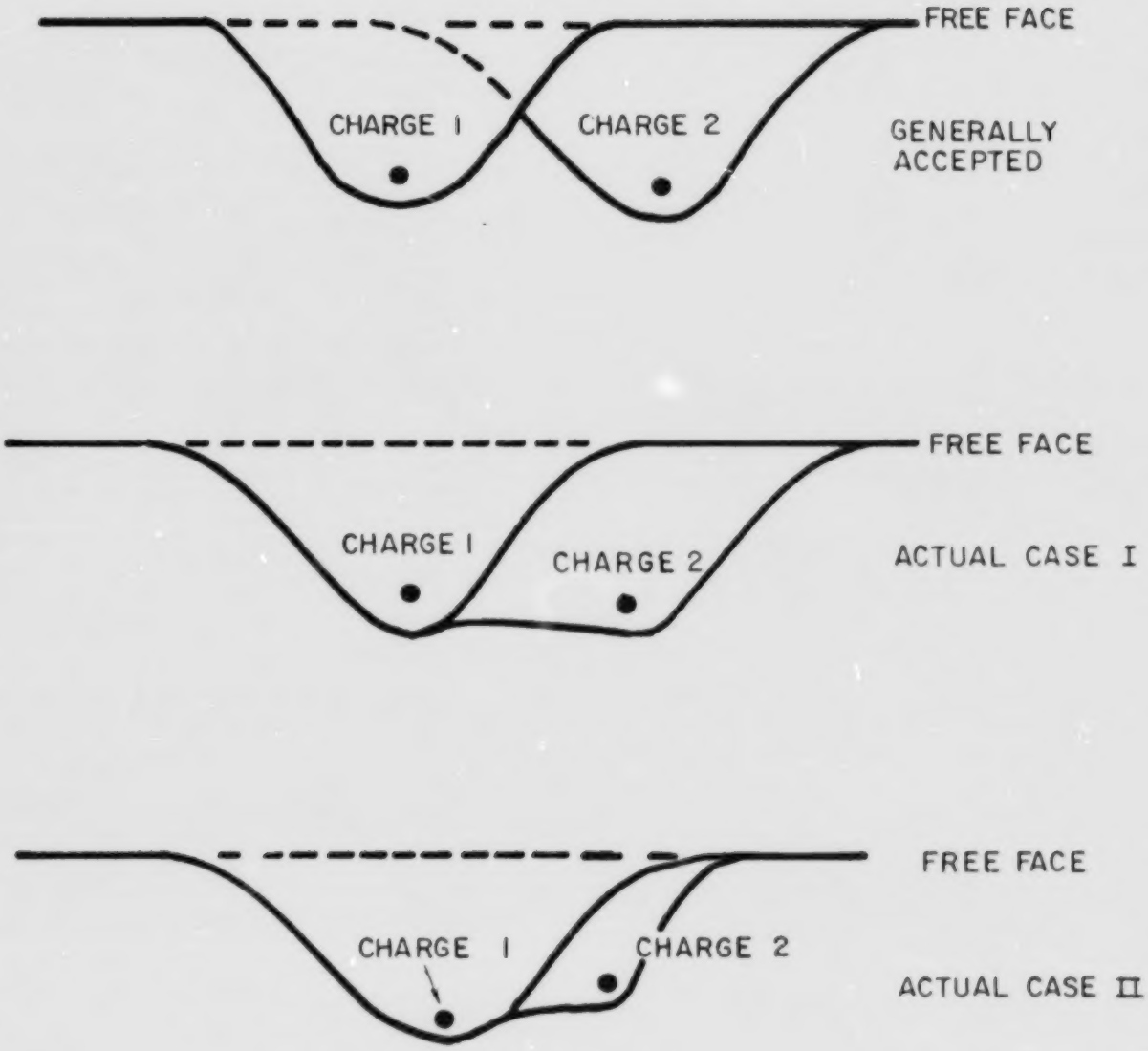


FIGURE 7 :
COMPARISON BETWEEN THE GENERALLY ACCEPTED DESIGN PRACTICE AND TWO ACTUAL CASES.

scaled crater radius - depth ratio curve. Therefore the maximum distance at which rock is broken can be found (distance AC in Figure 8) for every position of the charge. This distance which is called slant distance indicates the maximum distance which the charge is going to break due to its position with respect to the free faces surrounding it.

It is believed that this approach is reasonable since the closest face offering a relief to the blast determines the dimensions of the resulting crater.

4.4 Blast Design Flowchart

The program is split into two main parts as regards the flowchart. These parts correspond to the large diameter blasthole benching operation and the vertical crater retreat method. They will be dealt with separately in this section.

Large Diameter Blasthole Benching

Figures 9(A) and 9(B) present the general flowchart for the large diameter blasthole benching portion of the program.

After the initialization of the variables, the input data are read. Data can be read in either metric or imperial units. Output from data read in metric units is produced in metric units while imperial unit input data generates output in imperial units.

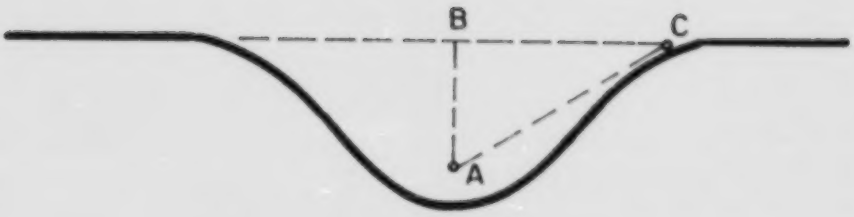


FIGURE 8 : MAXIMUM DISTANCE FROM A CHARGE IN WHICH ROCK IS BROKEN.

PROGRAM LDBB — GENERAL FLOWCHART

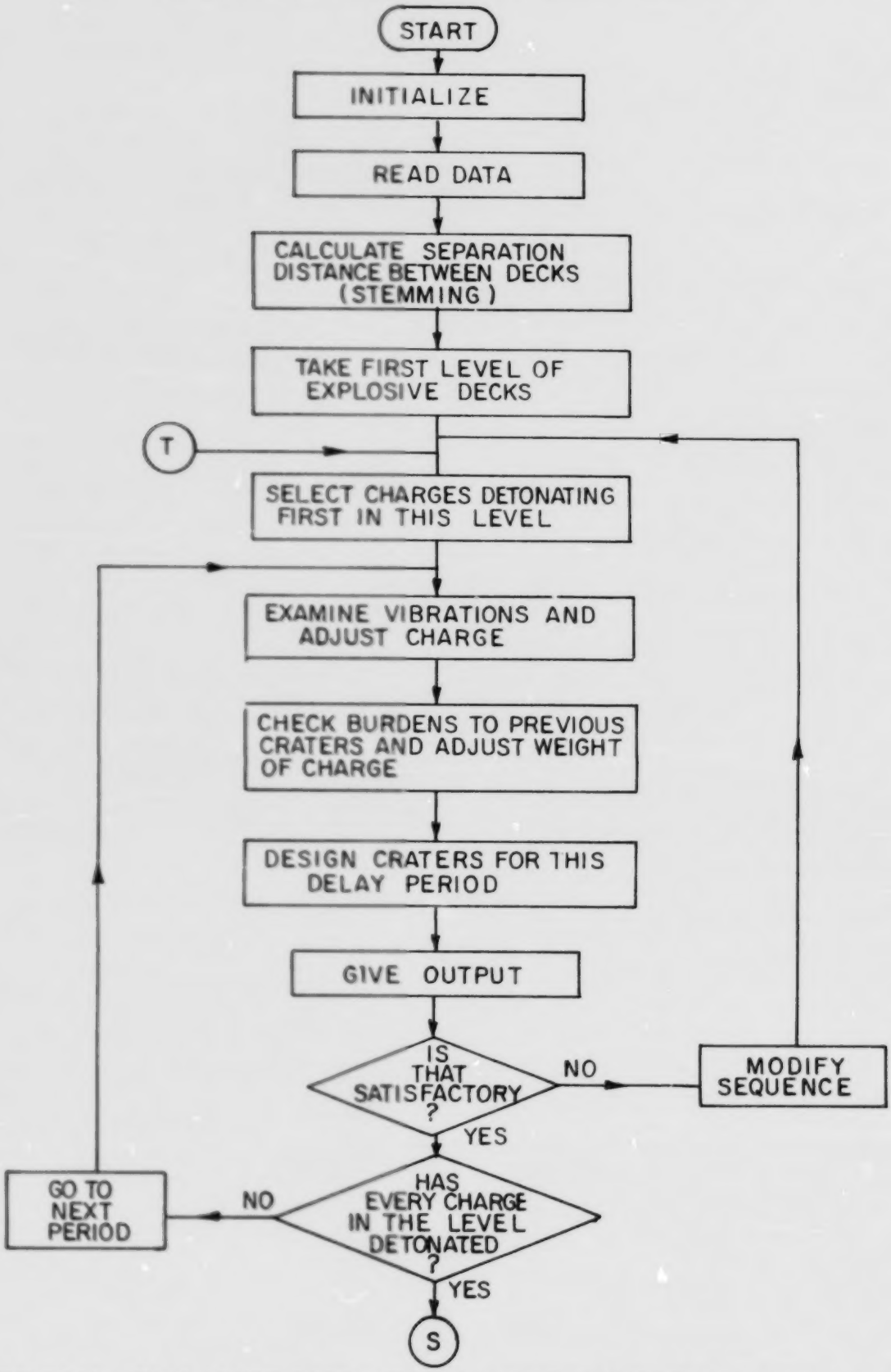


FIGURE 9(A): THE FIRST PART OF THE LARGE DIAMETER BLASTHOLE BENCH BLAST DESIGN PROGRAM FLOWCHART.

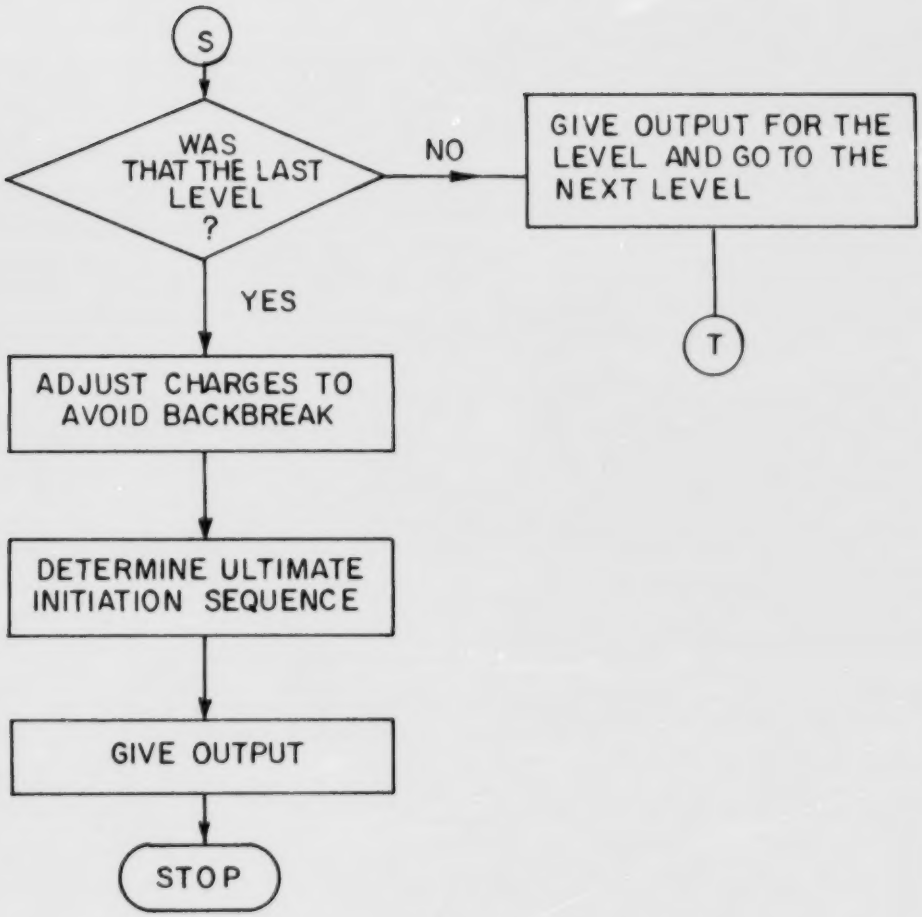


FIGURE 9(B) : THE SECOND PART OF THE LARGE DIAMETER BLASTHOLE BENCH BLAST DESIGN PROGRAM FLOWCHART.

The initial calculation of the separation between explosives decks is then made. Factors in the calculation include the rock type and other geological factors resulting in a required energy level for the material to produce good fragmentation. The influence of the different decks is accommodated by consideration of the fragmentation produced off the end of each deck charge, and in particular, the fragmentation produced by the end 8 charge diameters. This section of charge is considered separately as the shape of the stress wave form off the end of the charge is near spherical in shape and thus is typical of a spherical charge. The 8 charge diameter length is the longest charge that can be fired with results appearing to be similar to those that would be generated by an actual spherical charge itself.

Finally the deck separation is also considered in the light of the current experience which has shown the possibility of sympathetic detonation of delayed decks if the distance between them is too small. The best estimate available at the time of writing is that the deck separation should be no less than 16 times the hole diameter. This can be changed in the program later should a better relationship be established.

The next sequence of events is conducted on each successive charge and level of decks.

There are two methods of selecting the sequence of charge detonations. This may be inputted by the programmer or it may be generated by the program itself. The manual option has been introduced in the program to allow for overriding of the optimized self generated sequence should there be specific reasons for so doing by the mine engineer.

The charges are then examined for vibrations and adjustment is made if required. The burdens are checked to the previous craters and adjustments are made if necessary. The craters are then designed for these changes in the particular deck under consideration. The output for the process is then presented.

For the manual sequencing technique, the programmer is then asked if the output received to this point is satisfactory? if it is not the preceeding sequence of steps is repeated until the programmer is satisfied with the results. When the programmer is satisfied, or if the automatic mode is used, the program then checks to see if every charge in the level has been detonated. If they have not, again the preceeding sequence of events is repeated until all charges have been considered. Then the program proceeds.

The next question finds out if all of the different charge levels have been considered. If they have not the preceeding sequence is repeated for the next level, and so on, until all of the charges in the blast have been considered.

When all of the charges have been detonated the program then determines any special design with respect to wall control to avoid backbreak.

Finally the ultimate initiation sequence is determined and the output for the program is given.

Vertical Crater Retreat Method

Figure 10 presents the general flowchart for the vertical crater retreat method portion of the program.

Again the variables are initialized and the data are read. Again either metric or imperial units may be used.

A similar option to the one used in the bench blasting is available for the vertical crater retreat, this being the option to produce an optimized computer generated firing sequence for the charges, or to override this sequence and input a manual selection based on other factors.

The sequence of events in the vertical crater retreat design is now similar to a deck for the bench blasting program. The charge is examined for vibrations and adjusted if required. The burdens are then checked to previous craters and any adjustments in charge weight and delay are made. The new crater is then designed using, in this case, a three dimensional block model.

The program then checks if this is the last period to be detonated, if not the process is repeated. Once all of the charges have been detonated the blast is complete and the output is supplied.

PROGRAM VCRM - GENERAL FLOWCHART

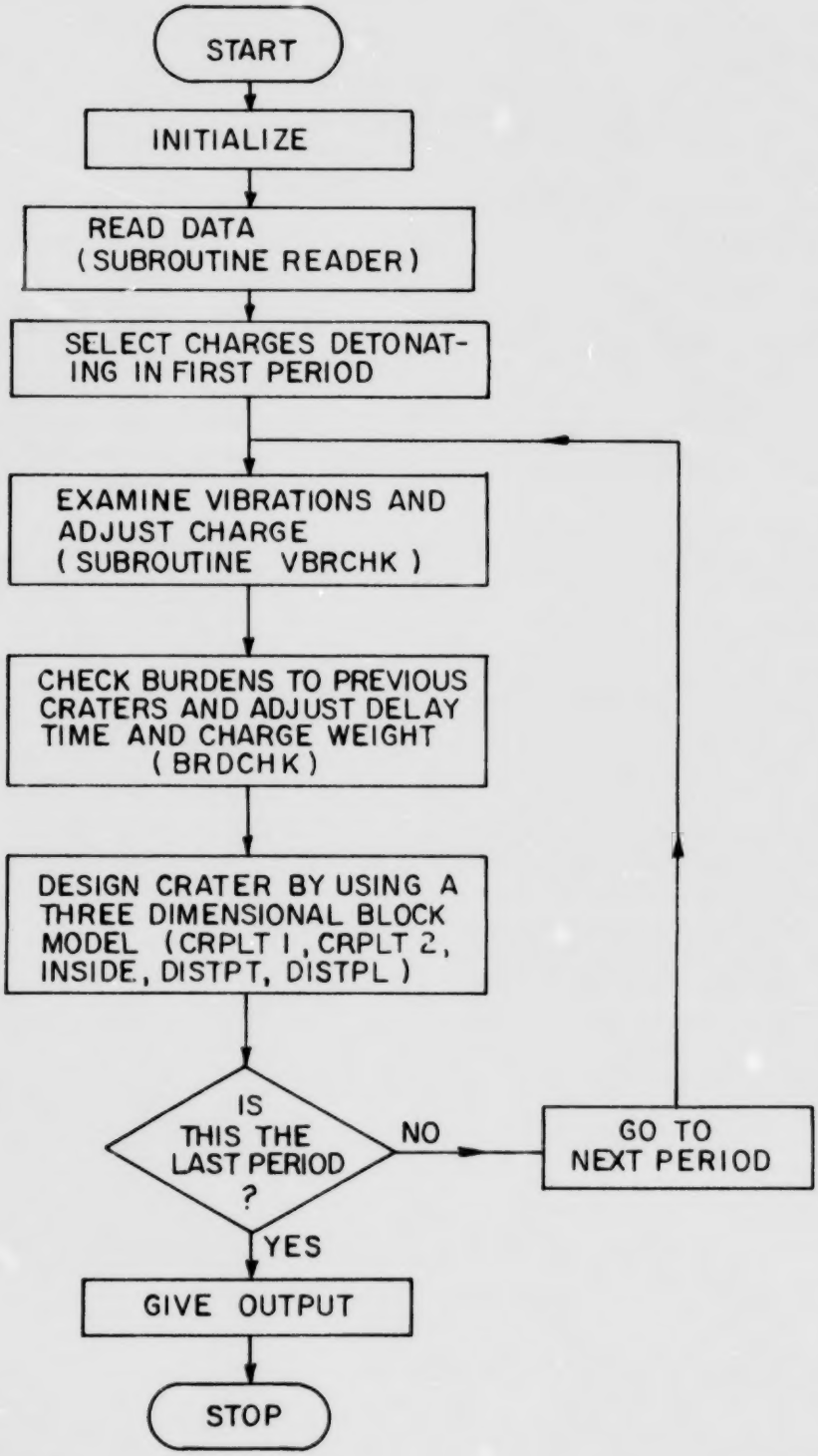


FIGURE 10: THE VERTICAL CRATER RETREAT METHOD BLAST DESIGN PROGRAM FLOWCHART.

Program Listing

A full listing of the program, split into four parts is presented in the Appendix A. The four parts are:

1. Vertical Crater Retreat Method Design
2. Large Diameter Bench Blast Design
3. Graphics Output Generator Program for Both Design Programs
4. Graphics Output Generator for the Vertical Crater Retreat after Blast Profile

4.5 Computer Software and Hardware

The programming language is VS FORTRAN which is compatible with most micro computers currently available.

The graphics used is the DI 3000 package prepared by Precision Visuals.

The computer used is the IBM 3081.

The graphics terminal used is the Tektronix 4107.

The plotter used is the Tektronix 4663S.

The program can be easily modified for use on most other currently available computers.

5.0 USERS MANUAL

This section shows how data are inputted into the program as well as presents the necessary exec files to run the program.

5.1 Input Data

The input data to the programs can be divided into four categories:

(a) Data related to the geometry of the blast. These data are:

- The coordinates of the holes
- The location of the free face
- The coordinates of the points in which vibrations will be checked.

(b) Data related to Livingston's crater theory. Data of this kind are:

- The scaled crater depth - depth ratio curve
- The scaled crater radius - depth ratio curve.

(c) Data related to the explosive. These data are:

- The weight strength of the explosive used in the tests and the explosive used in the blasting operations in the stope
- The density of the explosive.

(d) Other data such as:

- The diameter of the blastholes
- The number of the blastholes
- The site factors for the vibration checks
- The maximum acceptable levels of the peak particle velocity at different points.

Most of these data are fed in the program interactively. In the following it will be explained how to create the data files needed for the use of the program.

5.2 Data for the VCRM Program

The exec file (for the execution of the program) and the data files are shown in the Appendix B.

In the following the data files are described.

Unit 07: Data given interactively. The user has to respond to the questions of the program.

Unit 51: (INCOSTOP DATA) Data file containing the title and information regarding the geometry of the stope.

Line 1: Title (80 characters)

Line 2: FORMAT (3F10.2) x, y, z coordinates of reference point.

Line 3: FORMAT (3F10.2) stope length, stope width, stope depth.

LIne 4: FORMAT (F10.2) angle of local axis x with mine axis x.

Unit 61: (INCO61 DATA) Coordinates of the holes at several elevations.

Line 1: FORMAT (4I5, F10.2) Hole number, number of points along the hole at which measurements were taken, row number, column number, hole diameter.

Subsequent lines: FORMAT (3F10.2) x, y, z coordinate

After the last point the procedure is repeated.

The end of the file is marked with the number 99.

Unit 62: (PCH3 DATA) File containing information about the position of the charges in the boreholes. This file is created by the computer.

Unit 08: (CRCURV DATA) File generated interactively. It contains the data from the cratering curves. Once this file is created it does not have to be generated at each run of the program.

Unit 11: (VIBRAT DATA) File generated interactively. It contains data for the vibration checks.

Unit 70: (CHLOC DATA) Intermediate data file created by the computer.

Unit 77: (PLBLD DATA) Data file generated by the computer. It is used as data to the plotting program.

5.3 Data for Program LDBB

The exec file (for the execution of the program) and the data files are shown in the Appendix B. In the following a brief description of the data files will be given:

Unit 07: Data given interactively. The user has to respond to the programs questions.

Unit 08: (BCURV DATA) File generated interactively. It contains the data for the cratering curves. Once this file is created it can be used for the subsequent runs.

Unit 09: (BBLAST DATA) File containing geometrical data for the position of the free face.

Line 1: FORMAT (I5) Number of points defining the free face.

Subsequent Lines: FORMAT (2F10.2) x, y coordinates of each point.

Unit 11: (VIBRAT DATA) File generated interactively. It contains the data for the vibration checks. This file can be used in the subsequent runs.

Unit 12: (TOP DATA) File containing coordinates of points beyond which backbreak should not occur.

Line 1: FORMAT (I5) Number of segments defining the limit.

Subsequent Lines: FORMAT (4F10.2) x1, y1, x2, y2 coordinates of points defining each segment.

Unit 51: (INCOSTOP DATA) Data file containing the title and information regarding the geometry of the stope.

Line 1: FORMAT (A80) Title (80 characters)

Line 2: FORMAT (3F10.2) x, y, z coordinates of the reference point.

Line 3: FORMAT (3F10.2) stope length, stope width, depth.

Line 4: FORMAT (F10.2) angle of local axis x with mine axis x.

Unit 61: Intermediate data file created by the computer.

Unit 66: (INCO62 DATA) Coordinates of the holes at several elevations.

Line 1: FORMAT (4I5, F10.2) Hole number, number of points along the hole at which measurements were taken, row number, column number, hole diameter.

Line 2: FORMAT (3F10.2) x, y, z coordinates of the collar.

Subsequent Lines: FORMAT (3F10.2) x, y, z coordinates of the point. After the last point the procedure is repeated.

The end of the file is marked with number 99.

Unit 70: Intermediate file created by the computer.

Unit 62: (PCH3 DATA) File created by the computer. It contains the input data for the plotting program (PGRAPH).

Unit 72: (OUTLINE DATA) Data giving the outline of the stope.

Line 1: FORMAT (I5) Number of points giving outline.

Subsequent Lines: FORMAT (2F10.2) x, y coordinates of each point.

5.4 Data for Graphics Programs

Graphics output is given by using programs PGRAPH and PLBLAST. PGRAPH provides cross-sections and plan views at the blast block while PLBLAST gives the after blast profile of the VCR stope. The exec files for the use of these programs can be found in Appendix B. The input data for the programs are prepared by the programs used for the design of the blast (VCRM, LDBB). The user has to respond to a few questions asked by the programs.

6.0 TYPICAL PROGRAM OUTPUT

This section presents typical computer output for both the graphics and hard copy output for the blaster.

Figures 11 and 12 present the blasthole vertical and horizontal projections. This can be produced for any blasthole, whether the bench blasting or vertical crater retreat method is being used.

Figure 13 presents a typical stope plan. Plans can be produced for any level that may be desired.

Figures 14 through 17 present cross-sections of rows and columns of blastholes with the blasters information superimposed. The charge depth, charge size and required delay period are listed.

Figure 18 is a plan view of a VCR stope and Figures 19 and 20 are a typical row and column respectively with the blasthole loading information.

Appendix C presents typical output from the LDBB and VCR programs.

The large diameter blasthole bench output first presents the initiation sequence and explosive charge weights for each deck level. A plan view is provided of the broken material which is an aid for the manual operated initiation sequence only. It need not be reproduced if it is not desired.

A summary of the explosives required and number of the different delay periods follows. Finally the individual loading instructions for all of the blastholes is presented.

The vertical crater retreat output first presents the final modified blasthole initiation sequence, followed by the charge weight per hole. The blast fragmentation is illustrated in plan view either in a shaded form or using a digital presentation ready for contouring.

A summary of the explosives required and number of the different delay periods follows. Finally the individual loading instructions for all of the blastholes is presented.

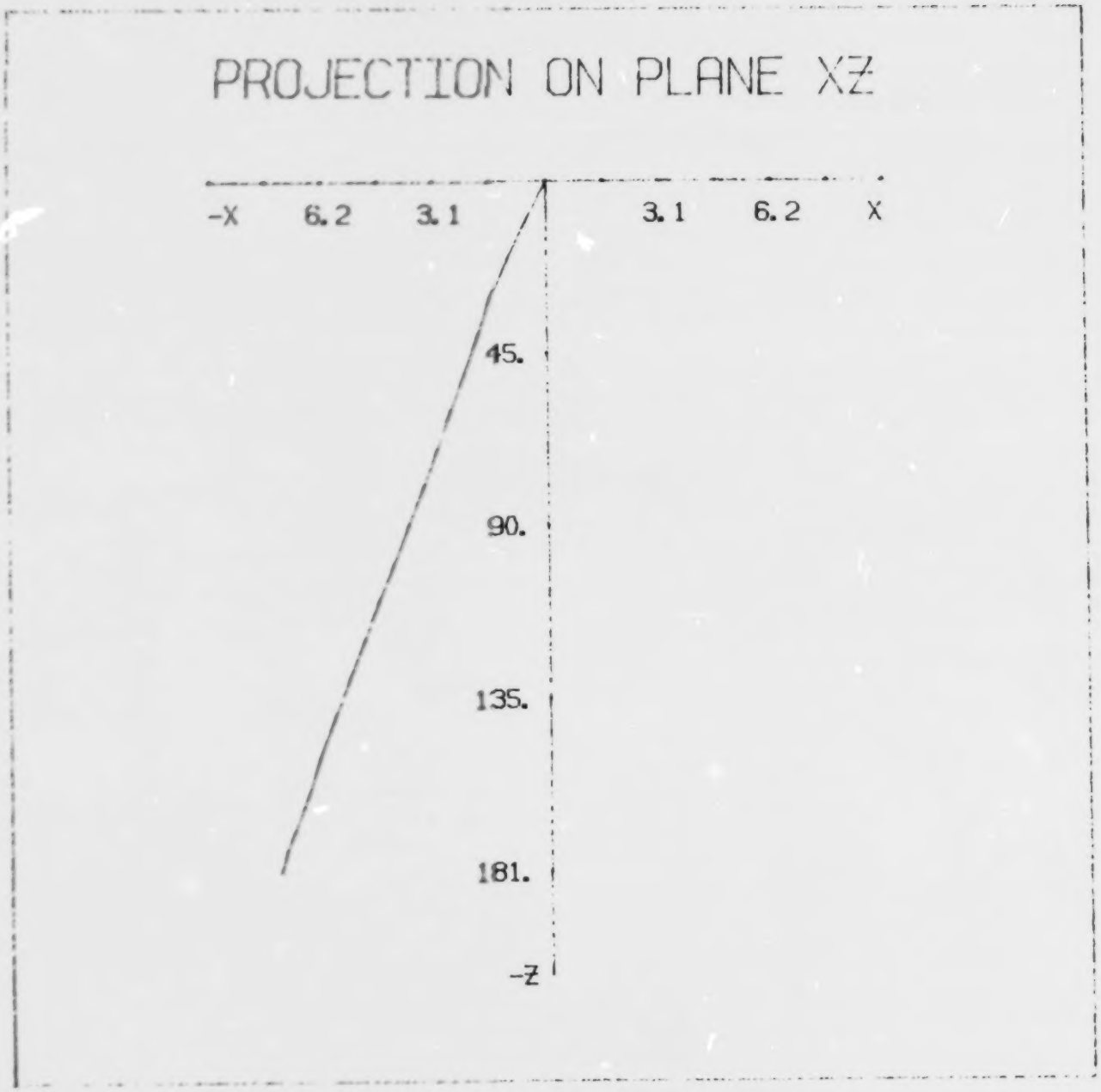


FIGURE 11: TYPICAL COMPUTER OUTPUT SHOWING BLASTHOLE PROJECTION ON PLANE XZ.

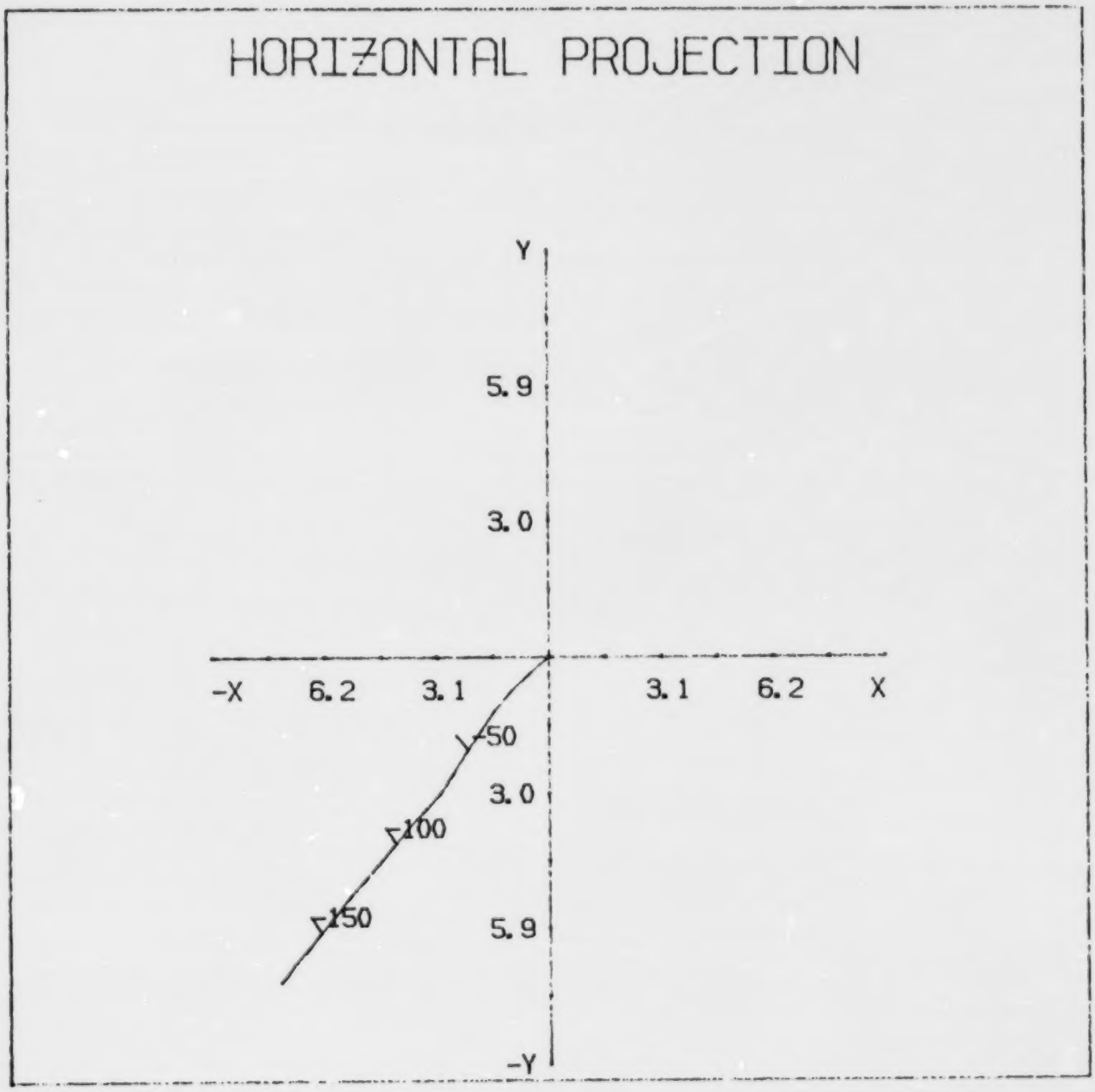


FIGURE 12: TYPICAL COMPUTER OUTPUT SHOWING HORIZONTAL BLASTHOLE PROJECTION.

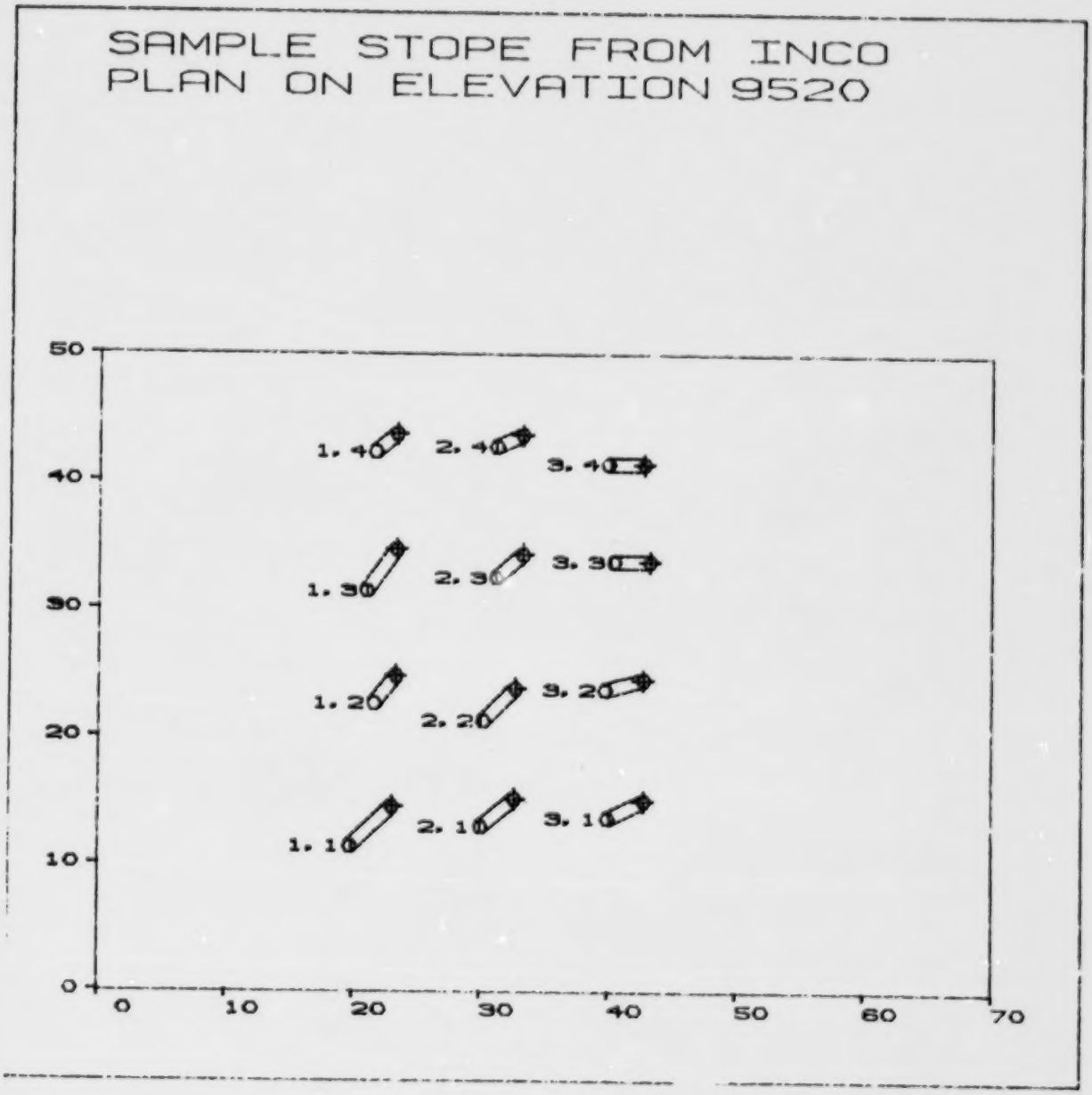


FIGURE 13: TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING PLAN VIEW OF A STOPE.

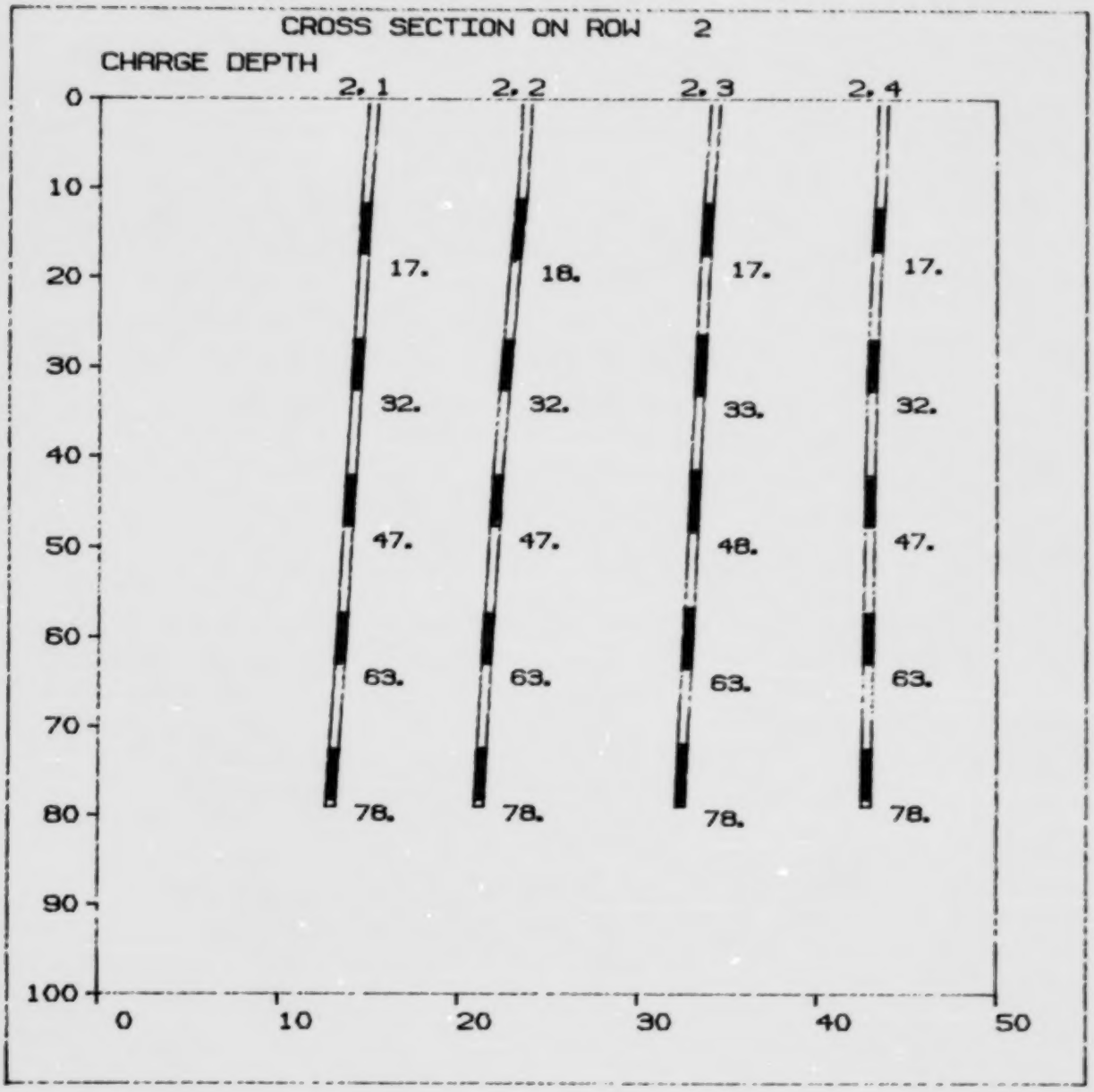


FIGURE 14: TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A ROW CROSS-SECTION WITH CHARGE DEPTHS.

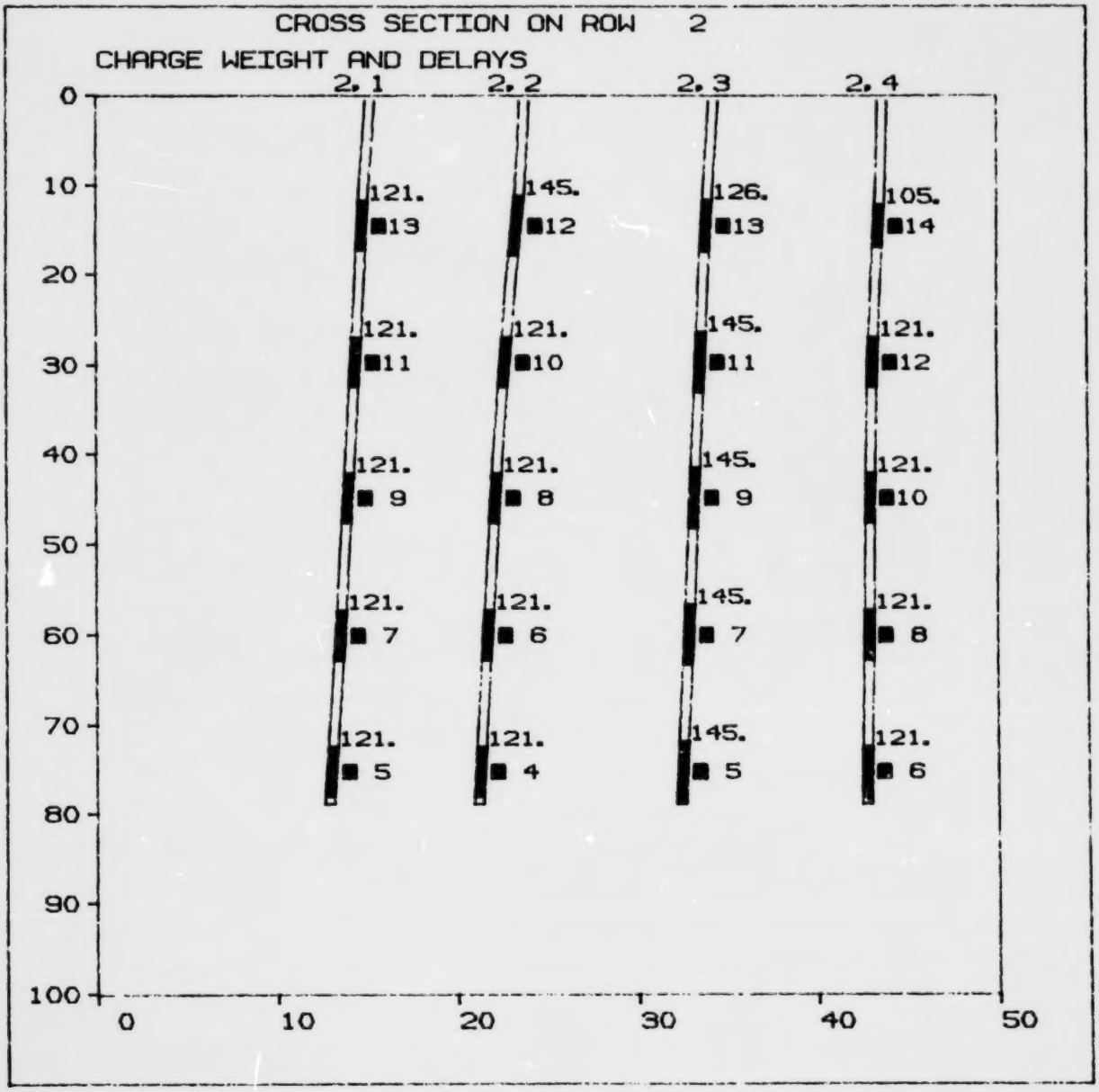


FIGURE 15: TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A ROW CROSS-SECTION WITH CHARGE WEIGHTS AND DELAYS.

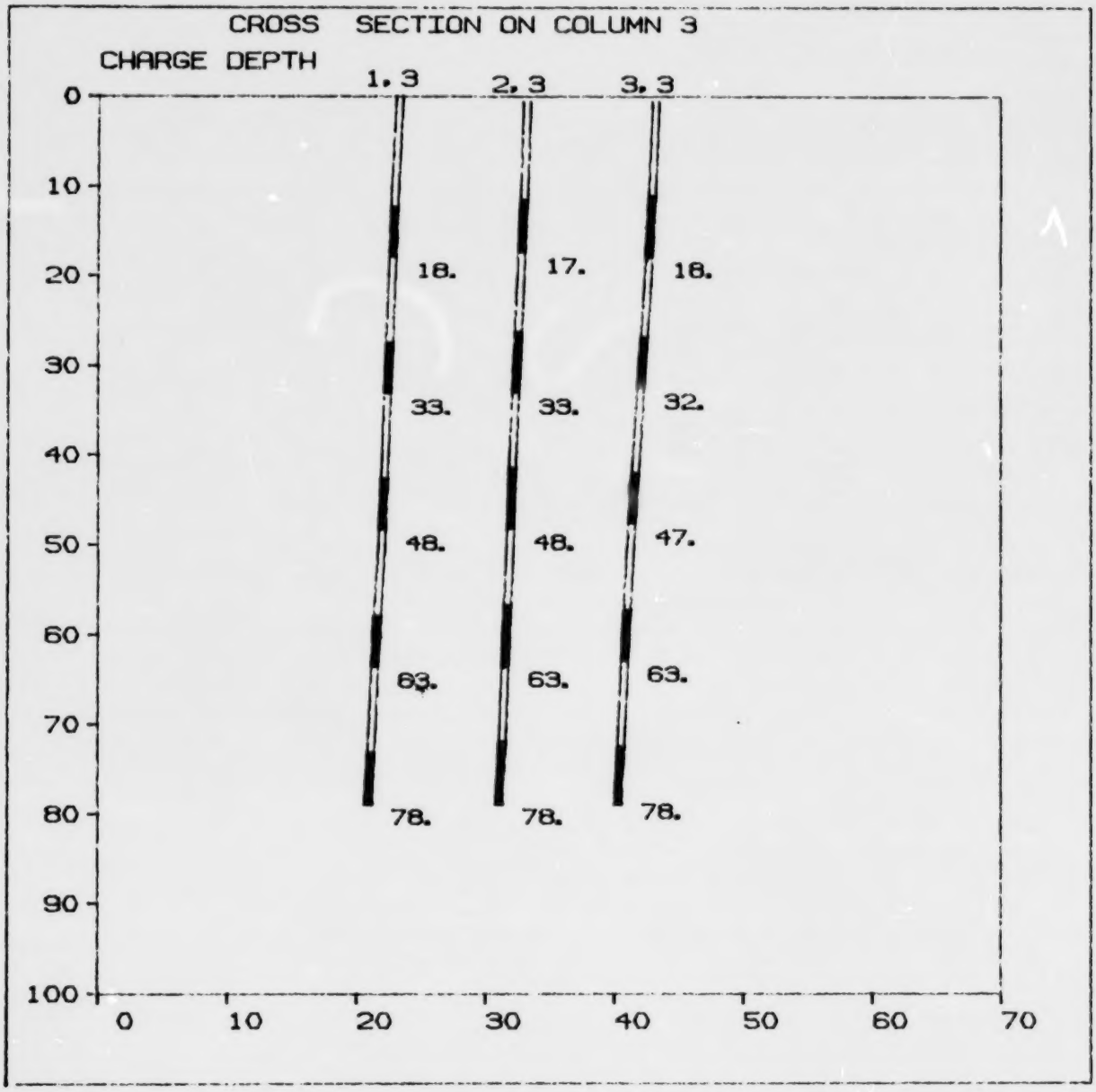


FIGURE 16: TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A COLUMN SECTION WITH CHARGE DEPTHS.

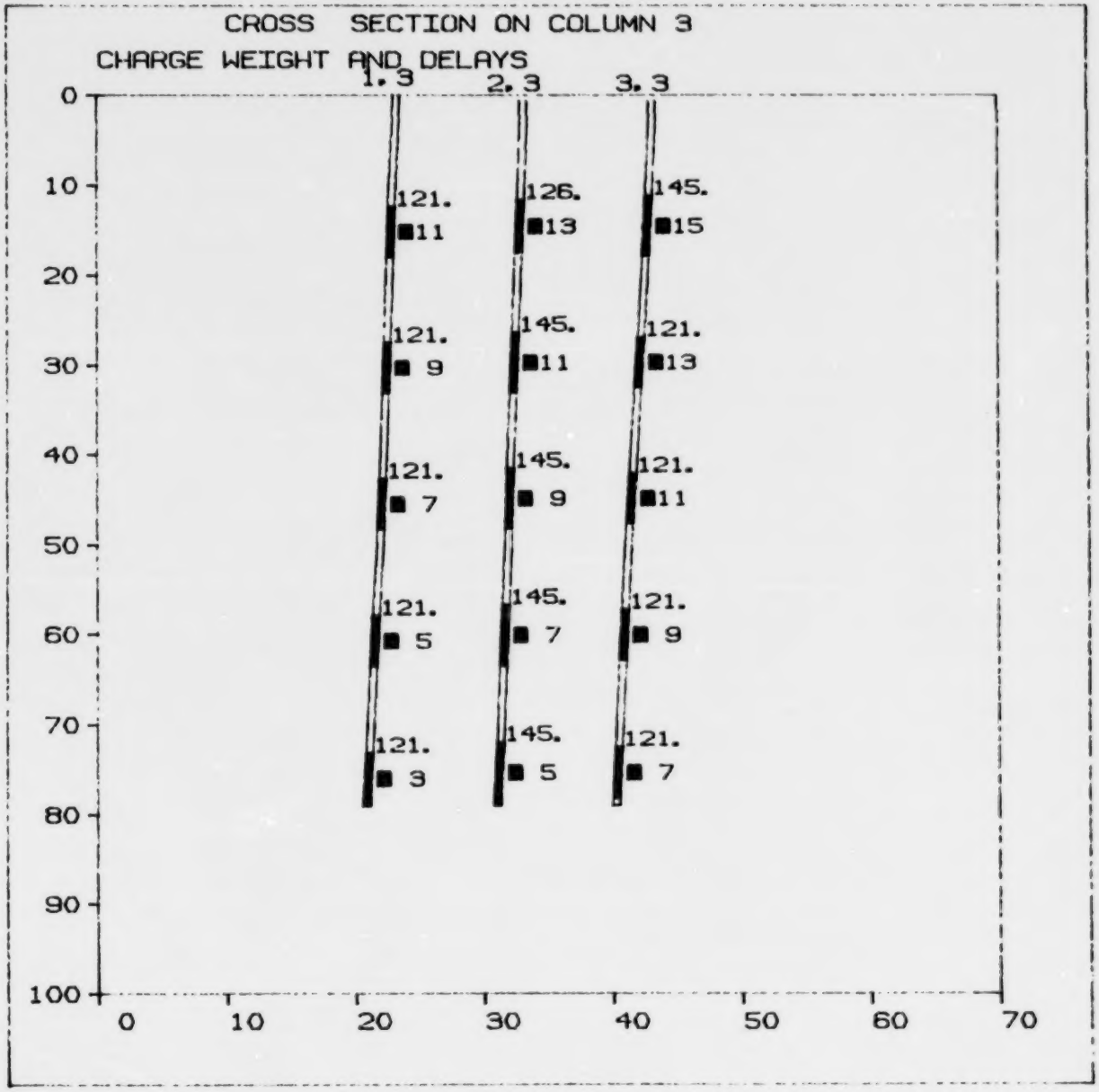


FIGURE 17: TYPICAL COMPUTER OUTPUT FOR THE LDBB PROGRAM SHOWING A COLUMN SECTION WITH CHARGE WEIGHTS AND DELAYS.

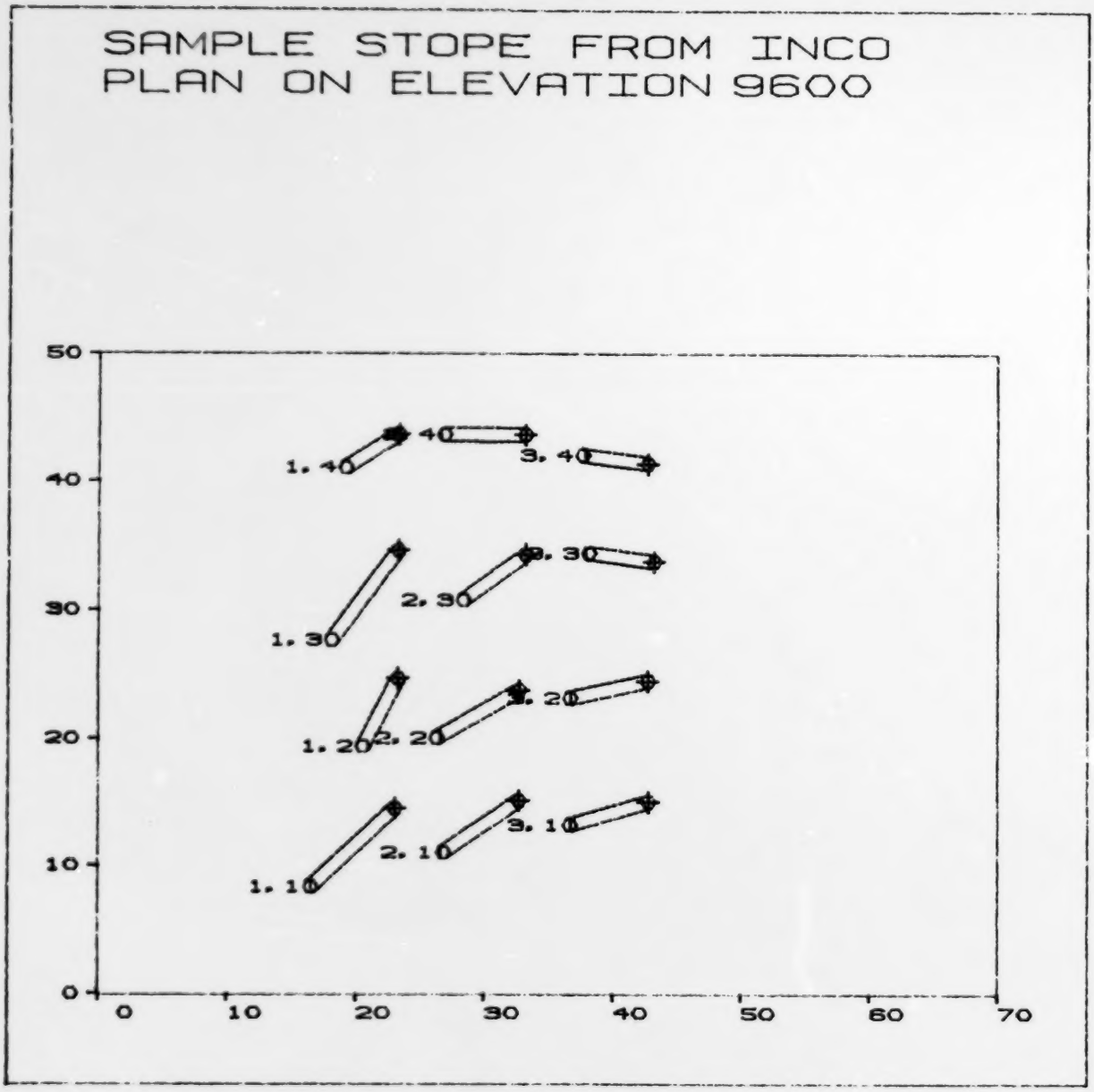


FIGURE 18: TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING STOPE PLAN.

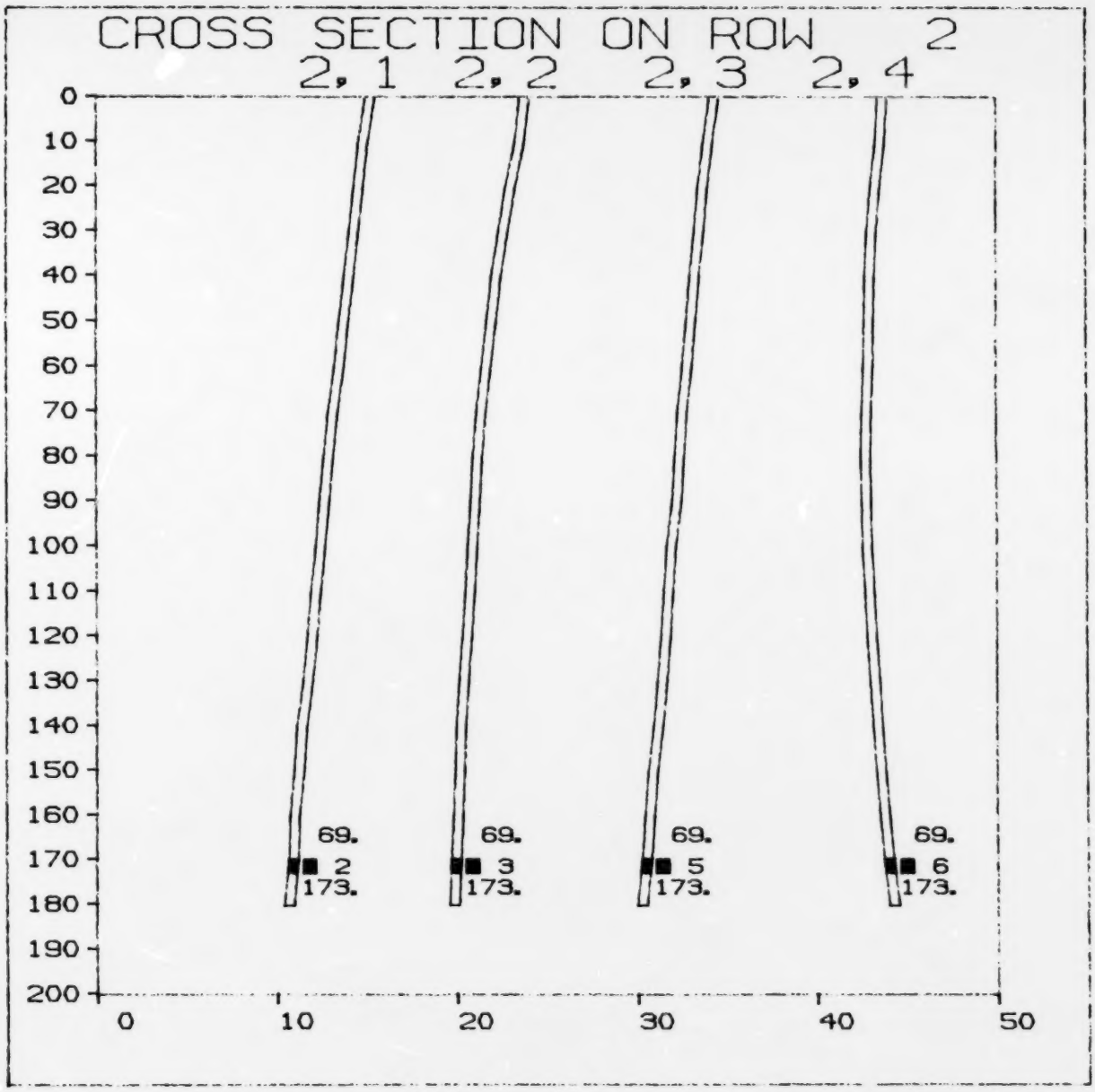


FIGURE 19: TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING A ROW SECTION WITH CHARGE WEIGHTS, DELAYS AND CHARGE DEPTHS.

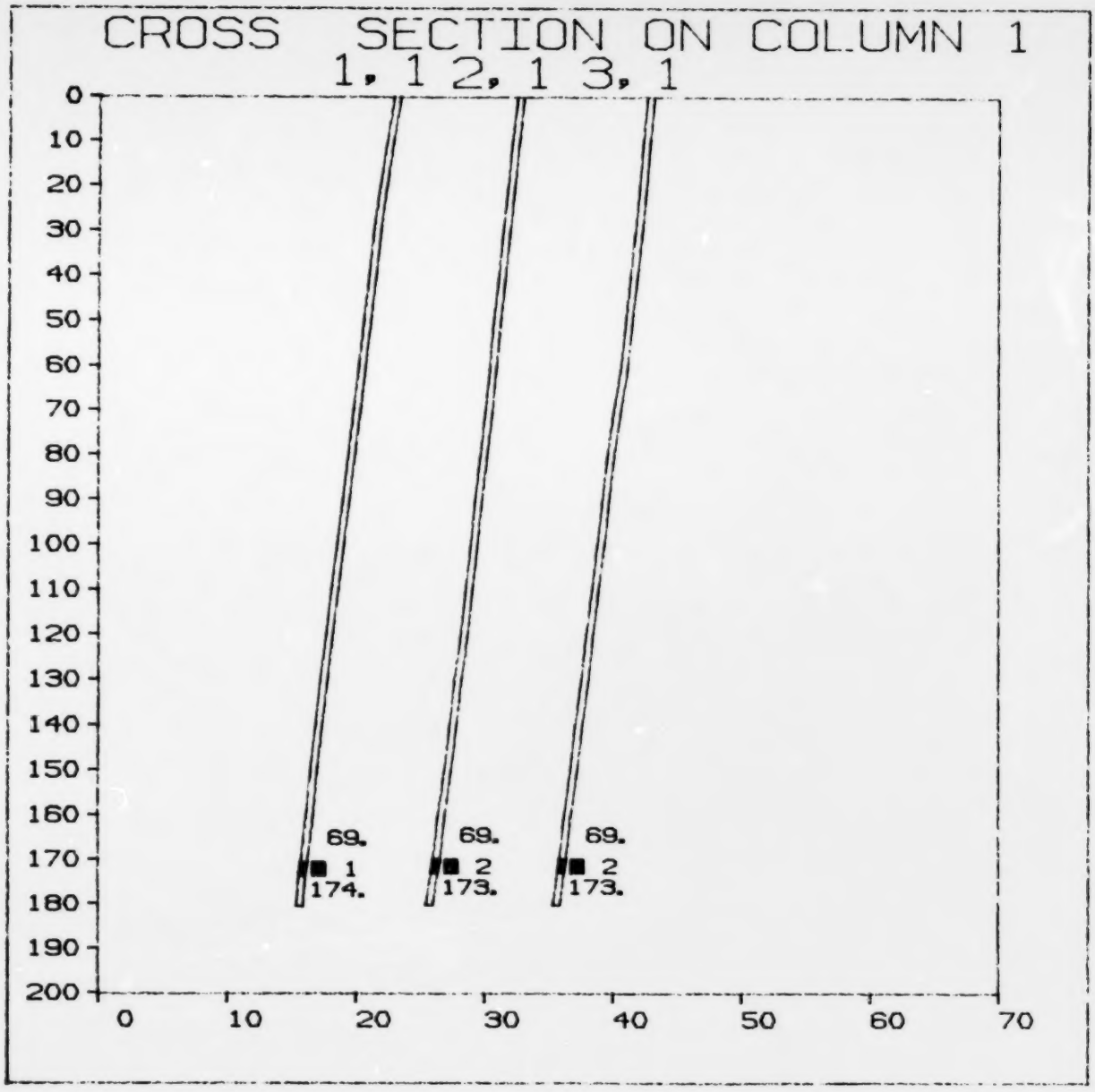


FIGURE 20: TYPICAL COMPUTER OUTPUT FOR THE VCRM PROGRAM SHOWING A COLUMN SECTION WITH CHARGE WEIGHTS, DELAYS AND CHARGE DEPTHS.

7.0 CONCLUSION

1. Two programs (VCRM and LDBB) have been developed to design and calculate a blast for a particular hole layout for VCR stopes, benches, circular and rectangular openings.

These programs consider:

- The effect of the multiple free faces on the blast.
- The effects of the delay sequencing on the blast.
- The properties of the rock and the explosive.
- The powder factor with respect to the desired fragmentation.
- The vibrations experienced in different points of the mine.
- The minimum deck separation distance to avoid sympathetic initiation.
- The minimization of backbreak.

2. The programs simulate the blast and give graphics output as well as a hard copy of the expected result.

3. The programs give instructions for the loading of the boreholes.

4. The programs provide a graphic model of the blast block.

BIBLIOGRAPHY

1. Aki, K., Fehler, M., Aamodt, R.L., Albright, J.N., Potter, R.M., Pearson, C.M., and Tester, J.W.

   1982: Interpretation of seismic data from hydraulic fracturing experiments at the Fenton Hill, New Mexico, Hot Dry Rock Geothermic Site. Journal of Geophysical Research, Vol. 87, p. 936-944.

2. Bauer, A., "Application of the Livingston Theory", Quarterly of the Colorado School of Mines Vol. 56 No. 1, 1961.

3. Devine, J.R., R.H. Beck, A.V.C. Meyer and W.I. Duval,

   1966: "Effect of Charge Weight on Vibration Levels from Quarry Blasting, USBM Report, Investigation 6774.

4. Lang, L.: "Vertical Crater Retreat: an Important New Mining Method" Underground Mining Methods Handbook SME - AIME pp 456 - 463.

5. Livingston, C.W., 1973: U.S. Patent 3,762,771, Oct. 2.

6. McCann, D.M., Grainger, P., and McCann

   1975: Inter-borehole acoustic measurements and their use in engineering geology. Geophysical Prospecting, Vol. 23, p. 50 - 69.

7. Roach R.: "VCR Modul" CIL internal report.

8. Wong, S., Hurley, P., and West, G.F.

   1983: Crosshole Seismology and Seismic imaging in crystalline rocks. Geophysical Research Letters, Vol, 10, p. 686 - 689.

APPENDIX A

COMPUTER PROGRAM PRINTOUTS

1. VERTICAL CRATER RETREAT METHOD DESIGN
2. LARGE DIAMETER BENCH BLAST DESIGN
3. GRAPHICS OUTPUT GENERATOR PROGRAM FOR BOTH DESIGN PROGRAMS
4. GRAPHICS OUTPUT GENERATOR FOR THE VERTICAL CRATER RETREAT AFTER BLAST PROFILE.

1. VERTICAL CRATER RETREAT METHOD DESIGN

```
      PROGRAM VCRM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C   THIS PROGRAM DESIGNS THE BLAST FOR A VCR STOPE WHEN THE POSITION C
C   OF THE HOLES AND THE CRATERING CHARACTERISTICS ARE KNOWN. THE    C
C   PROGRAM GIVES AN OUTPUT INDICATING PROBLEMS DUE TO THE DEVIATION C
C   OF THE HOLES                                                     C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       LAST EDITION :  NOVEMBER 1984
C
C
C     VARIABLES :
C            ALENG : LENGTH OF THE EXPLOSIVE COLUMN
C            DCONCH : DEPTH OF CONTAINED CHARGE (SPHERICAL)
C            DDOOPD : CHARGE LOCATION IN THE BOREHOLE
C            DELTA  : DEPTH RATIO
C            DELTAR : DEPTH RATIO
C            DELTO : OPTIMUM DEPTH RATIO
C            DEPTH  : DEPTH OF CRATER
C            DHOLE : DIAMETER OF BOREHOLE
C            DHOLEI : INCREASED DIAMETER OF BOREHOLE (FOR GRAPHICS)
C            DOPT(I) : DEPTH OF BURIAL
C            EFACTR : STRAIN ENERGY FACTOR FOR SPHERICAL CHARGES
C            ELEV : ELEVATION OF THE FREE FACE
C            EXPTYR : EXPLOSIVE TYPE
C            EXPTYP : EXPLOSIVE TYPE
C            IBLOCK(I,J,K) : IF GREATER THAN ZERO BLOCK (I.J) TAKEN
C                           OTHERWISE BLOCK IS NOT TAKEN
C            IPER(I) : DELAY PERIOD OF CHARGE I
C            ITIME(I.J.K) : ARRAY INDICATING PERIOD IN WHICH BLOCK (I.J)
C                  WAS REMOVED
C            IWR(I,J) : ARRAY FOR PLOTTING OF OUTPUT
C            KSYSUN  : 1 METRIC
C                      2 IMPERIAL
C            NP1     : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH
C                        RATIO CURVE (SPHERICAL)
C            NP2     : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH
C                        RATIO CURVE (SPHERICAL)
C
C            NUMBH  :  NUMBER OF HOLES IN THE BLAST
C            RHOE   : DENSITY OF THE EXPLOSIVE
C            SCDEPT : SCALED DEPTH OF BURIAL FOR SPHERICAL CHARGES
C            SCRAD : SCALED RADIUS FOR SPHERICAL
C            SCROP : OPTIMUM SCALED RADIUS
C            TITLE : TITLE
C            WEIGHT : WEIGHT OF EXPLOSIVE   CHARGE
C            WSTR  : WEIGHT STRENGTH
C            WTCHRG : WEIGHT OF CHARGE IN TESTS (SPHERICAL)
C            X(I) : X COORDINATE OF HOLE I
C            XB(I) : X COORDINATE OF POINT (BLOCK) (I.J,K)
C            XMAX : MAX LOCAL LENGTH (USED TO DETERMINE LENGTH OF BLOCK)
C            XMIN : MIN LOCAL LENGTH
C            Y(I) : Y COORDINATE OF HOLE I
```

```
C          YB(J) : Y COORDINATE OF POINT (BLOCK) (I,J,K)                VCR00560
C          YMAX : MAX LOCAL WIDTH (USED TO DETERMINE LENGTH OF BLOCK)   VCR00570
C          YMIN : MIN LOCAL WIDTH                                       VCR00580
C          Z(1) : Z COORDINATE OF HOLE I                                VCR00590
C          ZB(K) : Z COORDINATE OF POINT (BLOCK) (I,J,K)                VCR00600
C                                                                       VCR00610
C                                                                       VCR00620
      INTEGER*4 CLTIME,TCPU,LBS,INCH.FT                                 VCR00630
      CHARACTER*15 EXPTYP.EXPTYB                                        VCR00640
      CHARACTER*80 TITLE                                                VCR00650
      CHARACTER*8  QDATE ,QTIM                                          VCR00660
      CHARACTER*1 IFANCY(120)                                           VCR00670
      COMMON /RDRDT1/ DCONCH.WTCHRG.WSTRTH,SCDEPT(50).DELTA(50).        VCR00680
     XSCRAD(50),DHOLE,X(50),Y(50).Z(50).XVCHK(10),YVCHK(10),ZVCHK(10)   VCR00690
     X,PPVALL(10).AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),  VCR00700
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10).ZPIL3(10),PPVPAL.WSTR.    VCR00710
     XXMAX.XMIN,YMAX.YMIN,ZMAX.ZMIN,DELTAR(50),RHOE                     VCR00720
      COMMON/RDRDT2/ NP1,NP2,NUMBH.NCHK,NUMPIL.INIT(50),MSYSUN          VCR00730
      COMMON/IBLK/ IBLOCK(100.100,5),ITIME(100.100,5)                   VCR00740
      COMMON/DXDYDZ/ DX,DY.DZ                                           VCR00750
      COMMON/DELT/DELTO.EFACTR                                          VCR00760
      COMMON/XBLK/XB(100),YB(100),ZB(5)                                 VCR00770
      COMMON/ISEQ/ IPER(50).KK                                          VCR00780
      COMMON/FLY/ IFLY(50)                                              VCR00790
      COMMON/CHARGE/WEIGHT(50),DUPT(50),RCR(50),DEPTH(50).SCROP         VCR00800
      COMMON/BURDEN/ BURDEN(50,50)                                      VCR00810
      COMMON/TYPE/EXPTYP,EXPTYB                                         VCR00820
      COMMON/TT/TITLE                                                   VCR00830
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR.STLENG,STWIDT           VCR00840
     X ,STDEPT                                                          VCR00850
      COMMON/UNALP/ LBS.INCH.FT                                         VCR00860
      DIMENSION IWR(133,140).ZBOUT(5).ALENG(50).UDOUPD(50)              VCR00870
C-----------------------------------------------------------------------VCR00880
C  **      INITIALIZE DATA       **                                     VCR00890
C-----------------------------------------------------------------------VCR00900
      PI = 3.14159                                                      VCR00910
      DO 1901 I= 1,133                                                  VCR00920
      DO 1900 J= 1,140                                                  VCR00930
      IWR(I.J) = 0                                                      VCR00940
 1900 CONTINUE                                                          VCR00950
 1901 CONTINUE                                                          VCR00960
      DO 9000 I=1,50                                                    VCR00970
      INIT(I)=0                                                         VCR00980
 9000 CONTINUE                                                          VCR00990
C-----------------------------------------------------------------------VCR01000
C  **        READ DATA        **                                        VCR01010
C-----------------------------------------------------------------------VCR01020
      CALL READER                                                       VCR01030
C-----------------------------------------------------------------------VCR01040
C  **  FIND LIVINGSTON'S STRAIN ENERGY FACTOR  **                       VCR01050
C-----------------------------------------------------------------------VCR01060
      EFACTR = DCONCH / WTCHRG ** (1.0/3.0)                             VCR01070
C-----------------------------------------------------------------------VCR01080
C  **   FIND OPTIMUM DEPTH RATIO   **                                  VCR01090
C-----------------------------------------------------------------------VCR01100
```

```
      DO 10 I = 1, NP1                                                  VCR01110
      IF (SCDEPT(I+1).GE.SCDEPT(I)) GO TO 10                            VCR01120
      DELTO = DELTA(I)                                                  VCR01130
      IOPT = I                                                          VCR01140
      GO TO 14                                                          VCR01150
   10 CONTINUE                                                          VCR01160
   14 CONTINUE                                                          VCR01170
C-----------------------------------------------------------------------VCR01180
C  ** FIND SCALED RADIUS AT OPTIMUM DEPTH RATIO **                      VCR01190
C-----------------------------------------------------------------------VCR01200
      DO 15 I = 2, NP2                                                  VCR01210
      IF((DELTAR(I).GE.DELTO).AND.(DELTAR(I-1).LE.DELTO)) GO TO 16      VCR01220
   15 CONTINUE                                                          VCR01230
   16 SCROP = SCRAD(I-1) + (DELTO - DELTAR(I-1))/(DELTAR(I)-DELTAR(I-1))VCR01240
     X *(SCRAD(I)-SCRAD(I-1))                                           VCR01250
C-----------------------------------------------------------------------VCR01260
C  **     DETERMINE INITIATION SEQUENCE      **                         VCR01270
C-----------------------------------------------------------------------VCR01280
      CALL SEQUEN                                                       VCR01290
C-----------------------------------------------------------------------VCR01300
C  **    GET HOLE DETONATING IN PERIOD #1     **                        VCR01310
C-----------------------------------------------------------------------VCR01320
   11 DO 19 I = 1 ,NUMBH                                                VCR01330
      IF (IPER(I).GT.1) GO TO 19                                        VCR01340
      IFIRST = I                                                        VCR01350
      GO TO 20                                                          VCR01360
   19 CONTINUE                                                          VCR01370
   20 CONTINUE                                                          VCR01380
      IN = IFIRST                                                       VCR01390
      CALL VBRCHK (IFIRST)                                              VCR01400
C-----------------------------------------------------------------------VCR01410
C  **       GO TO NEXT PERIOD        **                                 VCR01420
C-----------------------------------------------------------------------VCR01430
      ID = 0                                                            VCR01440
  250 CONTINUE                                                          VCR01450
      DO 280 I = 1, NUMBH                                               VCR01460
      IF (IPER(I).NE.(IPER(IFIRST) + 1)) GO TO 280                      VCR01470
      IN = I                                                            VCR01480
      GO TO 281                                                         VCR01490
  280 CONTINUE                                                          VCR01500
  281 IFIRST = IN                                                       VCR01510
      CALL VBRCHK (IFIRST)                                              VCR01520
C-----------------------------------------------------------------------VCR01530
C  **  CHECK  FOR  MINIMUM  BURDENS  ( FLYROCK )   **                   VCR01540
C-----------------------------------------------------------------------VCR01550
      CALL BROCHK (IFIRST,KIND,INEXT)                                   VCR01560
      IF (KIND.NE.2) GO TO 400                                          VCR01570
      DO 390 I = 1, NUMBH                                               VCR01580
      IF(IPER(I).NE.IPER(INEXT)) GO TO 390                              VCR01590
      IN = I                                                            VCR01600
      GO TO 281                                                         VCR01610
  390 CONTINUE                                                          VCR01620
C-----------------------------------------------------------------------VCR01630
C  **  CHECK IF WE ARE IN THE LAST DELAY PERIOD  **                     VCR01640
C-----------------------------------------------------------------------VCR01650
```

```
  400 MAXPER = 0
      DO 425 III = 1 ,NUMBH
      IF (MAXPER.GT.IPER(III)) GO TO 425
      MAXPER = IPER(III)
  425 CONTINUE
      IF(IPER(IFIRST).LT.MAXPER ) GO TO 250
C-----------------------------------------------------------------------
C  **    D E S I G N    C R A T E R S    **
C-----------------------------------------------------------------------
      ZMAX = Z(1)
      ZMIN = 1000000.0
      DO 431 IT = 1,NUMBH
      ZZZ = ZMAX - DEPTH(IT)
      IF(ZMIN.LE.ZZZ) GO TO 431
      ZMIN = ZZZ
  431 CONTINUE
      CALL BLMOD( MAXPER)
C-----------------------------------------------------------------------
C  **  DESIGN CRATERS CREATED IN FIRST PERIOD  **
C-----------------------------------------------------------------------
      DO 450 ICR = 1, NUMBH
      IF(IPER(ICR).EQ.1) CALL CRPLT1 (ICR,1)
  450 CONTINUE
C-----------------------------------------------------------------------
C  **   CRATERS CREATED AFTER THE FIRST DELAY PERIOD  **
C-----------------------------------------------------------------------
      IPECR = 1
  475 IPECR = IPECR + 1
      DO 500  ICR = 1, NUMBH
      IF(IPER(ICR).NE.IPECR) GO TO 500
      CALL CRPLT2(ICR)
      CALL CRPLT1(ICR,IPECR)
  500 CONTINUE
      IF (IPECR.LT.MAXPER) GO TO 475
C-----------------------------------------------------------------------
C  ****        O  U  T  P  U  T          ****
C-----------------------------------------------------------------------
      WRITE(6,510)
  510 FORMAT(1H1,///////////,40X,'P  R  O  G  R  A  M    V  C  R  M  ',/
     X                    40X,' ---------------------------------',///.
     X 30X,'W R I T T E N   F O R   T H E   D E S I G N   O F   T H E',
     X //,30X,'B L A S T   I N   A N   U N D E R G R O U N D   S T O P '
     X,'E',//,30X,'U S I N G   T H E   V E R T I C A L   C R A T E R ',
     X//,30X,'R E T R E A T   M E T H O D . ',//)
      WRITE(6,511) TITLE
      CALL QTIME(QDATE,QTIM,CLTIME,TCPU)
  511 FORMAT(1H1,33X,A80,//)
      WRITE(6,690) QDATE,QTIM
  690 FORMAT(//////,15X,'DATE          :  ',A8,//15X,'TIME          :  ',A8)
      WRITE(6,512) EXPTYB
  512 FORMAT(1H0,'            EXPLOSIVE USED :',A15)
      WRITE(6,513) RHOE,LBS,INCH
  513 FORMAT(1H0,'            DENSITY        :',F6.4,2X,A4,'/CUBIC',A4)
      WRITE(6,590) DHOLE,INCH
  590 FORMAT(1H0,'            DIAMETER       :',F6.4,2X,A4)
```

```
      WRITE(6,496)
  496 FORMAT(' ',/////,51X,'MODIFIED INITIATION SEQUENCE',/
     X        51X,'-----------------------------',//
     X 40X,'HOLE NUMBER',30X,'DELAY PERIOD',/
     X 40X,'-----------',30X,'------------',//)
      DO 498  I=1,NUMBH
      WRITE(6,497)  I,IPER(I)
  497 FORMAT(45X,I2,38X,I2,/)
  498 CONTINUE
      WRITE(6,499) LBS
  499 FORMAT(' ',/////,51X,'WEIGHT OF CHARGES IN THE BLAST',/
     X        51X,'-----------------------------',//
     X  40X,'HOLE NUMBER',30X,'WEIGHT (',A4,')',/
     X  40X,'-----------',30X,'------------',//)
      DO 880 I = 1,NUMBH
      WRITE(6,881) I,WEIGHT(I)
  881 FORMAT(45X,I2,34X,F12.0)
  880 CONTINUE
      DO 710 I = 1,NUMBH
      ALENG(I) = WEIGHT(I)/(12.0*RHOEXP1*DHOLE**2.0/4.0)
      IF(MSYSUN.EQ.1) ALENG(I) = ALENG(I)*12.0/0.1
      Z(I) = Z(I) - DOPT(I)
  710 CONTINUE
      LONE = 1
      WRITE(70,701) NUMBH,LONE
  701 FORMAT(15I5)
      DHOLEI = 10*DHOLE
      WRITE(70,702) DHOLEI
  702 FORMAT(F10.2)
      WRITE(70,703) (Z(I),I=1,NUMBH)
      WRITE(70,703) (WEIGHT(I),I=1,NUMBH)
  703 FORMAT(8F10.2)
      WRITE(70,703) (ALENG(I),I=1,NUMBH)
      WRITE(70,704) (IPER(I),I=1,NUMBH)
  704 FORMAT(15I5)
      DO 1234 IOUT = 1,5
 1234 ZBOUT(IOUT) = ZB(IOUT)
      WRITE(6,436) ZMAX
  436 FORMAT(1H1,///////,' ',50X,'PLAN VIEW OF THE BLAST',/
     X            51X,'-----------------------',////
     X40X,'ELEVATIONS REPRESENTED BY THE PRINTOUT',/
     X40X,'-----------------------------------------',///
     X50X,'          ',F12.1,'  FT',//)
      WRITE(6,501)  ZBOUT(1)
  501 FORMAT(1H ,50X,'........',F12.1,'  FT',//)
      WRITE(6,502)  ZBOUT(2)
  502 FORMAT(1H ,50X,':::::::::',F12.1,'  FT',/
     X1H+,50X,'........',//)
      WRITE(6,503) ZBOUT(3)
  503 FORMAT(1H ,50X,'XXXXXXXXX',F12.1,'  FT',/
     X1H+,50X,':::::::::',/
     X1H+,50X,'........',//)
      WRITE(6,504) ZBOUT(4)
  504 FORMAT(1H ,50X,'00000000',F12.1,'  FT',/
     X1H+,50X,'XXXXXXXX',/
```

```
      X1H+,50X,'::::::::',/                                             VCR02760
      X1H+,50X,'........',//)                                           VCR02770
       WRITE(6,505) ZBOUT(5)                                            VCR02780
  505 FORMAT(1H ,50X,'AAAAAAAA',F12.1,'  IT',/                          VCR02790
      X1H+,50X,'########',/                                             VCR02800
      X1H+,50X,'00000000',/                                             VCR02810
      X1H+,50X,'XXXXXXXX',/                                             VCR02820
      X1H+,50X,'::::::::',/                                             VCR02830
      X1H+,50X,'........',///)                                          VCR02840
       WRITE(6,437) XB(1),XB(11),XB(21),XB(31),XB(41),XB(51),XB(61),    VCR02850
      XXB(71),XB(81),XB(91),XB(100)                                     VCR02860
  437 FORMAT(8X,11(F4.0,6X))                                            VCR02870
       WRITE(6,438)                                                     VCR02880
  438 FORMAT(10X,'+---------+---------+---------+---------+---------+', VCR02890
      X'---------+---------+---------+---------+---------+')            VCR02900
       DO 602 K = 1,5                                                   VCR02910
       DO 601 I = 1,100                                                 VCR02920
       DO 600 J = 1,100                                                 VCR02930
C      DO 550 K = 1, 5                                                  VCR02940
       IF (IBLOCK(I,J,K).EQ.0) GO TO 550                                VCR02950
       IWR(I,J) = K                                                     VCR02960
C      GO TO 600                                                        VCR02970
  550 CONTINUE                                                          VCR02980
  600 CONTINUE                                                          VCR02990
  601 CONTINUE                                                          VCR03000
  602 CONTINUE                                                          VCR03010
       DO 670 JP = 1,90,10                                              VCR03020
       DO 610 I = 1, 100                                                VCR03030
       IF(IWR(I,JP).NE.5) IFANCY(I)=' '                                 VCR03040
       IF(IWR(I,JP).EQ.5) IFANCY(I)='X'                                 VCR03050
  610 CONTINUE                                                          VCR03060
       WRITE(6,611) YB(JP),(IFANCY(I),I=1,100)                          VCR03070
       DO 620 I = 1, 100                                                VCR03080
       IF(IWR(I,JP).NE.3) IFANCY(I)=' '                                 VCR03090
       IF(IWR(I,JP).EQ.5) IFANCY(I)='#'                                 VCR03100
  620 CONTINUE                                                          VCR03110
       WRITE(6,612) (IFANCY(I),I=1,100)                                 VCR03120
       DO 613 I = 1,100                                                 VCR03130
       IF(IWR(I,JP).GE.4) IFANCY(I) = '0'                               VCR03140
       IF(IWR(I,JP).LT.4) IFANCY(I) = ' '                               VCR03150
  613 CONTINUE                                                          VCR03160
       WRITE(6,612) (IFANCY(I),I=1,100)                                 VCR03170
       DO 614 I = 1,100                                                 VCR03180
       IF(IWR(I,JP).GE.3) IFANCY(I) = 'X'                               VCR03190
       IF(IWR(I,JP).LT.3) IFANCY(I) = ' '                               VCR03200
  614 CONTINUE                                                          VCR03210
       WRITE(6,612) (IFANCY(I),I = 1,100)                               VCR03220
       DO 615 I = 1,100                                                 VCR03230
       IF(IWR(I,JP).GE.2) IFANCY(I) = ':'                               VCR03240
       IF(IWR(I,JP).LT.2) IFANCY(I) = ' '                               VCR03250
  615 CONTINUE                                                          VCR03260
       WRITE(6,612) (IFANCY(I),I= 1,100)                                VCR03270
       DO 616 I = 1,100                                                 VCR03280
       IF(IWR(I,JP).GE.1) IFANCY(I) = '.'                               VCR03290
       IF(IWR(I,JP).LT.1) IFANCY(I) = ' '                               VCR03300
```

```
  616 CONTINUE                                                          VCR03310
      WRITE(6,612) (IFANCY(I),I = 1,100)                                VCR03320
  611 FORMAT(1H ,F4.0,'    + ',100A1)                                   VCR03330
  612 FORMAT(1H+,4X  ,'      ',100A1)                                   VCR03340
      JF=JP+1                                                           VCR03350
      JL=JP+9                                                           VCR03360
      DO 660 JJP = JF,JL                                                VCR03370
      DO 640 I = 1, 120                                                 VCR03380
      IF(IWR(I,JJP).NE.5) IFANCY(I)=' '                                 VCR03390
      IF(IWR(I,JJP).EQ.5) IFANCY(I)='A'                                 VCR03400
  640 CONTINUE                                                          VCR03410
      WRITE(6,641) (IFANCY(I),I=1,100)                                  VCR03420
      DO 694 I = 1, 100                                                 VCR03430
      IF(IWR(I,JJP).NE.5) IFANCY(I)=' '                                 VCR03440
      IF(IWR(I,JJP).EQ.5) IFANCY(I)='*'                                 VCR03450
  694 CONTINUE                                                          VCR03460
      WRITE(6,642) (IFANCY(I),I=1,100)                                  VCR03470
      DO 643 I = 1,100                                                  VCR03480
      IF(IWR(I,JJP).GE.4) IFANCY(I) = 'O'                               VCR03490
      IF(IWR(I,JJP).LT.4) IFANCY(I) = ' '                               VCR03500
  643 CONTINUE                                                          VCR03510
      WRITE(6,642) (IFANCY(I),I=1,100)                                  VCR03520
      DO 644 I = 1,100                                                  VCR03530
      IF(IWR(I,JJP).GE.3) IFANCY(I) = 'X'                               VCR03540
      IF(IWR(I,JJP).LT.3) IFANCY(I) = ' '                               VCR03550
  644 CONTINUE                                                          VCR03560
      WRITE(6,642) (IFANCY(I),I = 1,100)                                VCR03570
      DO 645 I = 1,100                                                  VCR03580
      IF(IWR(I,JJP).GE.2) IFANCY(I) = ':'                               VCR03590
      IF(IWR(I,JJP).LT.2) IFANCY(I) = ' '                               VCR03600
  645 CONTINUE                                                          VCR03610
      WRITE(6,642) (IFANCY(I),I= 1,100)                                 VCR03620
      DO 646 I = 1,100                                                  VCR03630
      IF(IWR(I,JJP).GE.1) IFANCY(I) = '.'                               VCR03640
      IF(IWR(I,JJP).LT.1) IFANCY(I) = ' '                               VCR03650
  646 CONTINUE                                                          VCR03660
      WRITE(6,642) (IFANCY(I),I = 1,100)                                VCR03670
  641 FORMAT(1H ,4X  ,'    | ',100A1)                                   VCR03680
  642 FORMAT(1H+,4X  ,'      ',100A1)                                   VCR03690
  660 CONTINUE                                                          VCR03700
  670 CONTINUE                                                          VCR03710
      WRITE(6,9436) ZMAX,(ZBOUT(IOUT),IOUT=1,5)                         VCR03720
 9436 FORMAT(1H1,///////,' ',50X,'PLAN VIEW OF THE BLAST',/             VCR03730
     X            51X,'----------------------',////                    VCR03740
     X40X,'ELEVATIONS REPRESENTED BY THE NUMBERS OF THE PRINTOUT',/     VCR03750
     X40X,'-------------------------------------------------------',///VCR03760
     X50X,'NUMBER 0',F12.1,'  FT',//                                    VCR03770
     X50X,'NUMBER 1',F12.1,'  FT',//                                    VCR03780
     X50X,'NUMBER 2',F12.1,'  FT',//                                    VCR03790
     X50X,'NUMBER 3',F12.1,'  FT',//                                    VCR03800
     X50X,'NUMBER 4',F12.1,'  FT',//                                    VCR03810
     X50X,'NUMBER 5',F12.1,'  FT',///)                                  VCR03820
      WRITE(6,9437) XB(1),XB(11),XB(21),XB(31),XB(41),XB(51),XB(61),    VCR03830
     XXB(71),XB(81),XB(91),XB(100)                                      VCR03840
 9437 FORMAT(8X,11(F4.0,6X))                                            VCR03850
```

```
      WRITE(6,9438)                                                     VCR03860
 9438 FORMAT(10X,'+---------+---------+---------+---------+---------+', VCR03870
     X'---------+---------+---------+---------+---------+')            VCR03880
      DO 9602 K = 1,5                                                   VCR03890
      DO 9601 I = 1,100                                                 VCR03900
      DO 9600 J = 1,100                                                 VCR03910
C     DO 550 K = 1, 5                                                   VCR03920
      IF (IBLOCK(I,J,K).EQ.0) GO TO 9550                                VCR03930
      IWR(I,J) = K                                                      VCR03940
C     GO TO 600                                                         VCR03950
 9550 CONTINUE                                                          VCR03960
 9600 CONTINUE                                                          VCR03970
 9601 CONTINUE                                                          VCR03980
 9602 CONTINUE                                                          VCR03990
      DO 9670 JP = 1,90,10                                              VCR04000
      WRITE(6,9655) YB(JP),(IWR(I,JP),I=1,100)                          VCR04010
 9655 FORMAT(1H ,F4.0,'   + ',100I1)                                    VCR04020
      WRITE(6,9660) ((IWR(I,J),I=1,100),J=JP+1,JP+9)                    VCR04030
 9660 FORMAT(1H ,4X  ,'   | ',100I1)                                    VCR04040
 9670 CONTINUE                                                          VCR04050
      DO 9995 J = 1,100                                                 VCR04060
      WRITE(77,7777) (IWR(I,J),I=1,100)                                 VCR04070
C     WRITE(77,777) (IWR(I,J),J=81,100)                                 VCR04080
C 777 FORMAT(20I1)                                                      VCR04090
 7777 FORMAT(80I1)                                                      VCR04100
 9995 CONTINUE                                                          VCR04110
C-----------------------------------------------------------------------VCR04120
C  **    GIVE SUMMARY OF THE REQUIRED SUPPLIES    **                    VCR04130
C-----------------------------------------------------------------------VCR04140
      TW = 0.0                                                          VCR04150
      DO 9100 I = 1,NUMBH                                               VCR04160
      TW = TW + WEIGHT(I)                                               VCR04170
 9100 CONTINUE                                                          VCR04180
      WRITE(6,9101) TW,LBS                                              VCR04190
 9101 FORMAT(1H1,45X,'SUMMARY',/45X,'-------',///,25X,'TOTAL WEIGHT OF' VCR04200
     X,' EXPLOSIVE  :',F10.0,2X,A4)                                     VCR04210
      WRITE(6,9102)                                                     VCR04220
 9102 FORMAT(1H0,25X,'DELAY NUMBER',40X,'QUANTITY',/                    VCR04230
     X           ,25X,'------------',40X,'--------')                    VCR04240
      MAXPER = 0                                                        VCR04250
      DO 9105 I = 1,NUMBH                                               VCR04260
      IF(IPER(I).GT.MAXPER) MAXPER=IPER(I)                              VCR04270
 9105 CONTINUE                                                          VCR04280
      DO 9110 I = 1,MAXPER                                              VCR04290
      MM = 0                                                            VCR04300
      DO 9109 J = 1,NUMBH                                               VCR04310
      IF(IPER(J).EQ.I) MM = MM + 1                                      VCR04320
 9109 CONTINUE                                                          VCR04330
      WRITE(6,9115) I,MM                                                VCR04340
 9115 FORMAT(1H0,30X,I2,48X,I2)                                         VCR04350
 9110 CONTINUE                                                          VCR04360
      CALL CHARGL(DBOUPD)                                               VCR04370
      WRITE(6,9200) FT,LBS                                              VCR04380
 9200 FORMAT(1H1,45X,'INSTRUCTIONS FOR LOADING',/                       VCR04390
     X           ,45X,'----------------------',///                      VCR04400
```

```
     X10X,'HOLE NUMBER',10X,'CHARGE LOCATION',10X,'CHARGE WEIGHT',10X,  VCR04410
     X'DELAY NUMBER',/10X,'           ',10X,'    (',A4,')    ',14X,'(', VCR04420
     XA4,')',/10X,'---------------',10X,'-----------------',10X,        VCR04430
     X'-------------',10X,'------------')                               VCR04440
      DO 9260 I = 1,NUMBH                                               VCR04450
      WRITE(6,9250) I,DDOUPD(I),WEIGHT(I),IPER(I)                       VCR04460
 9250 FORMAT(1H0,15X,I2,19X,F5.0,19X,F5.0,17X,I2)                       VCR04470
 9260 CONTINUE                                                          VCR04480
 1000 STOP                                                              VCR04490
      END                                                               VCR04500
      SUBROUTINE READER                                                 VCR04510
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCVCR04520
C                                                                      CVCR04530
C   PURPOSE : TO READ GENERAL DATA FOR THE DESIGN OF THE VCR BLAST     CVCR04540
C   THE DATA ARE READ FROM THE FOLLOWING UNITS  :                      CVCR04550
C   UNIT  7 - TERMINAL ( INTERACTIVELY )                               CVCR04560
C   UNIT  8 - DISK FILE WITH THE DATA FOR THE CRATERING CURVES         CVCR04570
C   UNIT  9 - DISK FILE WITH DATA FOR THE STOPE                        CVCR04580
C   UNIT 10 - INITIATION SEQUENCE                                      CVCR04590
C   UNIT 11 - DISK FILE WITH DATA FOR THE VIBRATION CHECKS. THIS FILE  CVCR04600
C             IS CREATED BY THE COMPUTER IN A PREVIOUS RUN.            CVCR04610
C                                                                      CVCR04620
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCVCR04630
C                                                                       VCR04640
C    VARIABLES :                                                        VCR04650
C          DCONCH : DEPTH OF CONTAINED CHARGE (SPHERICAL)               VCR04660
C          DDOUPD : CHARGE LOCATION IN THE BOREHOLE                     VCR04670
C          DELTA  : DEPTH RATIO                                         VCR04680
C          DELTAR : DEPTH RATIO                                         VCR04690
C          DHOLE : DIAMETER OF BOREHOLE                                 VCR04700
C          ELEV : ELEVATION OF THE FREE FACE                            VCR04710
C          EXPTYB : EXPLOSIVE TYPE                                      VCR04720
C          EXPTYP : EXPLOSIVE TYPE                                      VCR04730
C          IPER(I) : DELAY PERIOD OF CHARGE I                           VCR04740
C          MSYSUN  : 1 METRIC                                           VCR04750
C                    2 IMPERIAL                                         VCR04760
C          NP1    : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH     VCR04770
C                     RATIO CURVE (SPHERICAL)                           VCR04780
C          NP2    : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH    VCR04790
C                     RATIO CURVE (SPHERICAL)                           VCR04800
C                                                                       VCR04810
C          NUMBH  :  NUMBER OF HOLES IN THE BLAST                       VCR04820
C          RHOE   : DENSITY OF THE EXPLOSIVE                            VCR04830
C          SCDEPT : SCALED DEPTH OF BURIAL FOR SPHERICAL CHARGES        VCR04840
C          SCRAD : SCALED RADIUS FOR SPHERICAL                          VCR04850
C          SCROP : OPTIMUM SCALED RADIUS                                VCR04860
C          TITLE : TITLE                                                VCR04870
C          X(I) : X COORDINATE OF HOLE I                                VCR04880
C          XMAX : MAX LOCAL LENGTH (USED TO DETERMINE LENGTH OF BLOCK)  VCR04890
C          XMIN : MIN LOCAL LENGTH                                      VCR04900
C          XPIL1,XPIL2,XPIL3 : X COORD OF POINTS DEFINING PILLAR        VCR04910
C          Y(I) : Y COORDINATE OF HOLE I                                VCR04920
C          YMAX : MAX LOCAL WIDTH (USED TO DETERMINE LENGTH OF BLOCK)   VCR04930
C          YMIN : MIN LOCAL WIDTH                                       VCR04940
C          YPIL1,YPIL2,YPIL3 : Y COORD OF POINTS DEFINING PILLAR        VCR04950
```

```
C          Z(I) : Z COORDINATE OF HOLE I                                VCR04960
C          ZMAX : MAX LOCAL DEPTH (USED TO DETERMINE LENGTH OF BLOCK)   VCR04970
C          YMIN : MIN LOCAL DEPTH                                       VCR04980
C          ZPIL1,ZPIL2,ZPIL3 : Z COORD OF POINTS DEFINING PILLAR        VCR04990
C                                                                       VCR05000
C                                                                       VCR05010
C                                                                       VCR05020
C                                                                       VCR05030
C                                                                       VCR05040
      CHARACTER*1 KODE, KODE1                                           VCR05050
      CHARACTER*4 LBS,INCH,FT                                           VCR05060
      CHARACTER*15 EXPTYP,EXPTYB                                        VCR05070
      CHARACTER*80 TITLE                                                VCR05080
C-----------------------------------------------------------------------VCR05090
C   **        INCLUDE COMMONS          **                               VCR05100
C-----------------------------------------------------------------------VCR05110
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),        VCR05120
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)   VCR05130
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),  VCR05140
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,    VCR05150
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE                     VCR05160
      COMMON/RDRDT2/ NP1,NP2,NUMBH,NCHK,NUMPIL,INIT(50),MSYSUN          VCR05170
      COMMON/ISEQ/ IPER(50),KK                                          VCR05180
      COMMON/CHARGE/ WEIGHT(50),DOPT(50),RCR(50),DEPTH(50),SCROP        VCR05190
      COMMON/TYPE/EXPTYP,EXPTYB                                         VCR05200
      COMMON/TT/TITLE                                                   VCR05210
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT           VCR05220
     X ,STDEPT                                                          VCR05230
      COMMON/UNALF/ LBS,INCH,FT                                         VCR05240
C-----------------------------------------------------------------------VCR05250
C   **  READ CRATERING CHARACTERISTICS   **                             VCR05260
C-----------------------------------------------------------------------VCR05270
      NUMPIL=0                                                          VCR05280
      WRITE (7,9)                                                       VCR05290
    9 FORMAT('  ENTER TYPE OF EXPLOSIVE USED IN THE TESTS :')           VCR05300
      READ (7,1009) EXPTYP                                              VCR05310
 1009 FORMAT(A15)                                                       VCR05320
      WRITE (7,1010)                                                    VCR05330
 1010 FORMAT(' ENTER SYSTEM OF UNITS YOU ARE USING :',/                 VCR05340
     X'      (1) M E T R I C',/                                         VCR05350
     X'      (2) I M P E R I A L ')                                     VCR05360
      READ(7,*) MSYSUN                                                  VCR05370
      IF (MSYSUN.EQ.1) GO TO 1020                                       VCR05380
      LBS=' LBS'                                                        VCR05390
      FT = ' FT '                                                       VCR05400
      INCH ='INCH'                                                      VCR05410
      GO TO 1030                                                        VCR05420
 1020 LBS = ' KG '                                                      VCR05430
      FT = ' M  '                                                       VCR05440
      INCH = ' CM '                                                     VCR05450
 1030 WRITE (7,10) FT,LBS                                               VCR05460
   10 FORMAT ('  ENTER DEPTH OF CONTAINED CHARGE (',A4,'),',/           VCR05470
     X         '          WEIGHT OF EXPLOSIVE (',A4,')')               VCR05480
      READ (7,*) DCONCH, WTCHRG                                         VCR05490
      WRITE (7,11)                                                      VCR05500
```

```
   11 FORMAT ('  WHAT IS THE WEIGHT STRENGTH OF THE EXPLOSIVE',/
     X        '  RELATIVE TO AN/FO ?')
      READ (7,*) WSTRTH
C-----------------------------------------------------------------------
C  **        READ CRATERING CURVES          **
C-----------------------------------------------------------------------
      WRITE (7,12)
   12 FORMAT ('  DO YOU WANT TO INPUT NEW CRATERING CURVES  ? ')
      READ (7,13) KODE
   13 FORMAT (1A1)
      IF (KODE.EQ.'N') GO TO 30
      WRITE (7,14)
   14 FORMAT ('  ENTER NUMBER OF POINTS TO DESIGN THE ',/
     X        '                SCALED CRATER DEPTH - DEPTH RATIO',/
     X        '                CURVE : ')
      READ (7,*) NP1
      WRITE (8,15) NP1
   15 FORMAT (I5)
      WRITE (7,16)
   16 FORMAT('  ENTER SCALED DEPTH, DEPTH RATIO ',/
     X        '                FOR EACH POINT OF THE CURVE.')
      DO 19 I = 1, NP1
      READ (7,*) SCDEPT(I), DELTA(I)
      WRITE(8,18) SCDEPT(I), DELTA(I)
   18 FORMAT(2F10.4)
   19 CONTINUE
      WRITE(7,20)
   20 FORMAT('  ENTER NUMBER OF POINTS TO DESIGN ',/
     X        '                THE SCALED CRATER RADIUS - DEPTH RATIO',/
     X        '                CURVE.')
      READ (7,*) NP2
      WRITE(8,15) NP2
      WRITE(7,26)
   26 FORMAT('  ENTER SCALED CRATER RADIUS, DEPTH RATIO',/
     X        '                FOR EACH POINT OF THE CURVE.')
      DO 29 I = 1, NP2
      READ (7,*) SCRAD(I), DELTAR(I)
      WRITE(8,18) SCRAD(I), DELTAR(I)
   29 CONTINUE
      GO TO 40
C-----------------------------------------------------------------------
C  **  READ CURVES FROM FILE  **
C-----------------------------------------------------------------------
   30 READ (8,15) NP1
      READ (8,18) ((SCDEPT(I), DELTA(I)), I=1,NP1)
      READ (8,15) NP2
      READ (8,18) ((SCRAD(I), DELTAR(I)), I=1,NP2)
   40 CONTINUE
C-----------------------------------------------------------------------
C   **   READ DATA OF THE STOPE  **
C-----------------------------------------------------------------------
      WRITE(7,201) INCH
  201 FORMAT ('  ENTER THE DIAMETER OF THE BLASTHOLES (',A4,') :')
      READ (7,*) DHOLE
      WRITE(7,202)
```

```
  202 FORMAT('  ENTER NUMBER OF HOLES IN THE BLAST :')
      READ (7,*) NUMBH
C-----------------------------------------------------------------------
C   **  READ X,Y COORDINATES OF THE HOLES AT A PARTICULAR LEVEL Z  **
C-----------------------------------------------------------------------
      WRITE(7,209)
  209 FORMAT('  ENTER THE ELEVATION OF THE FREE FACE OF THE BLAST :')
      READ(7,*) ELEV
      REWIND 51
      READ(51,519) TITLE
      READ(51,520) XREFER,YREFER,ZREFER
      READ(51,520) STLENG,STWIDT,STDEPT
      READ(51,520) STANGR
      STANGR=3.14159*STANGR/180.0
  519 FORMAT(A80)
  520 FORMAT(4F10.2)
      CALL PLVIEW(ELEV,X,Y,Z,NUMBH)
      WRITE(9,210) ((X(I),Y(I),Z(I)), I = 1, NUMBH)
  210 FORMAT (3F15.5)
C-----------------------------------------------------------------------
C  **    READ DATA FOR VIBRATION CHECKS       **
C     SPECIFY POINTS OF PARTICULAR INTEREST AND ACCEPTABLE
C     VIBRATIONS IN THESE POINTS.
C-----------------------------------------------------------------------
      WRITE (7,250)
  250 FORMAT('     THE FOLLOWING DATA ARE FOR CHECKING VIBRATION'./
     X        '     LEVELS AT PILLARS  AND OTHER PARTICULAR POINTS'./
     X        '     OF INTEREST ....'./
     X        '     ARE THESE NEW OR PREVIOUS DATA ?')
      READ (7,13) KODE
      IF (KODE.EQ.'P') GO TO 490
      WRITE(7,270)
  270 FORMAT('    ENTER THE NUMBER OF POINTS IN WHICH VIBRATIONS ARE'./
     X        '    TO BE CHECKED:')
      READ (7,*) NCHK
      IF(NCHK.EQ.0) GO TO  331
      WRITE(11,280)
  280 FORMAT(I5)
      WRITE(7,300)
  300 FORMAT('    ENTER SPECIFIC POINTS IN WHICH VIBRATIONS WILL BE'.
     X        ' CHECKED.',/
     X'  ( EACH POINT MUST BE SPECIFIED BY ITS COORDINATES SEPARATED',/
     X'    BY COMMAS. AND THE MAXIMUM ACCEPTABLE LEVEL OF PEAK PARTICLE'
     X,/'     VELOCITY. E.G. 12.4,12.6,14.2,10.0)')
      DO 315 I = 1, NCHK
      READ (7,*) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)
C-----------------------------------------------------------------------
C  **  CONVERT TO LOCAL COORDINATES   **
C-----------------------------------------------------------------------
      XVCHK(I) = XVCHK(I) - XREFER
      YVCHK(I) = YVCHK(I) - YREFER
      ZVCHK(I) = ZVCHK(I) - ZREFER
      XC = XVCHK(I)*COS(STANGR) + YVCHK(I)*SIN(STANGR)
      YC = -XVCHK(I)*SIN(STANGR) + YVCHK(I)*COS(STANGR)
      XVCHK(I) = XC
```

```
      YVCHK(I) = YC                                                     VCR06610
      WRITE(11,310) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)             VCR06620
  310 FORMAT(4F15.5)                                                    VCR06630
  315 CONTINUE                                                          VCR06640
      WRITE(7,320)                                                      VCR06650
  320 FORMAT('  ENTER THE SITE FACTORS K, M FOR THE CALCULATION OF THE'.VCR06660
     X' PEAK PARTICLE',/'                VELOCITY :')                   VCR06670
      READ (7,A) AKFAC, AMFAC                                           VCR06680
      WRITE(11,330) AKFAC, AMFAC                                        VCR06690
  330 FORMAT (2F15.5)                                                   VCR06700
  331 CONTINUE                                                          VCR06710
C-----------------------------------------------------------------------VCR06720
C  AA        ENTER DATA FOR PILLARS        AA                           VCR06730
C-----------------------------------------------------------------------VCR06740
      WRITE (7,430)                                                     VCR06750
  430 FORMAT('  ARE THERE ANY PILLARS IN WHICH YOU WOULD LIKE TO CHECK 'VCR06760
     X,'VIBRATIONS ?')                                                  VCR06770
      READ (7,431) KODE1                                                VCR06780
  431 FORMAT (1A1)                                                      VCR06790
      IF (KODE1.EQ.'N') GO TO 490                                       VCR06800
      WRITE(7,440)                                                      VCR06810
  440 FORMAT('  ENTER NUMBER OF PILLARS :')                             VCR06820
      READ (7,A) NUMPIL                                                 VCR06830
      WRITE(11,450) NUMPIL                                              VCR06840
  450 FORMAT(I5)                                                        VCR06850
      DO 480 I = 1, NUMPIL                                              VCR06860
      WRITE(7,460) I                                                    VCR06870
  460 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 1ST POINT OF PILLAR'. VCR06880
     X' NO.',I3)                                                        VCR06890
      READ (7,A) XPIL1(I), YPIL1(I), ZPIL1(I)                           VCR06900
C-----------------------------------------------------------------------VCR06910
C  AA  CONVERT TO LOCAL COORDINATES   AA                                VCR06920
C-----------------------------------------------------------------------VCR06930
      XPIL1(I) = XPIL1(I) - XREFER                                      VCR06940
      YPIL1(I) = YPIL1(I) - YREFER                                      VCR06950
      ZPIL1(I) = ZPIL1(I) - ZREFER                                      VCR06960
      XC = XPIL1(I)ACOS(STANGR) + YPIL1(I)ASIN(STANGR)                  VCR06970
      YC = -XPIL1(I)ASIN(STANGR) + YPIL1(I)ACOS(STANGR)                 VCR06980
      XPIL1(I) = XC                                                     VCR06990
      YPIL1(I) = YC                                                     VCR07000
      WRITE(11,470) XPIL1(I), YPIL1(I), ZPIL1(I)                        VCR07010
  470 FORMAT (3F15.5)                                                   VCR07020
      WRITE(7,471) I                                                    VCR07030
  471 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 2ND POINT OF PILLAR', VCR07040
     X' NO.', I3)                                                       VCR07050
      READ (7,A) XPIL2(I), YPIL2(I), YPIL2(I)                           VCR07060
C-----------------------------------------------------------------------VCR07070
C  AA  CONVERT TO LOCAL COORDINATES   AA                                VCR07080
C-----------------------------------------------------------------------VCR07090
      XPIL2(I) = XPIL2(I) - XREFER                                      VCR07100
      YPIL2(I) = YPIL2(I) - YREFER                                      VCR07110
      ZPIL2(I) = ZPIL2(I) - ZREFER                                      VCR07120
      XC = XPIL2(I)ACOS(STANGR) + YPIL2(I)ASIN(STANGR)                  VCR07130
      YC = -XPIL2(I)ASIN(STANGR) + YPIL2(I)ACOS(STANGR)                 VCR07140
      XPIL2(I) = XC                                                     VCR07150
```

```
      YPIL2(I) = YC                                                     VCR07160
      WRITE(11,470) XPIL2(I), YPIL2(I), ZPIL2(I)                        VCR07170
      WRITE(7,473) I                                                    VCR07180
  473 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 3RD POINT OF PILLAR', VCR07190
     X' NO.', I3)                                                       VCR07200
      READ (7,A) XPIL3(I), YPIL3(I), ZPIL3(I)                           VCR07210
C-----------------------------------------------------------------------VCR07220
C  AA  CONVERT TO LOCAL COORDINATES   AA                                VCR07230
C-----------------------------------------------------------------------VCR07240
      XPIL3(I) = XPIL3(I) - XREFER                                      VCR07250
      YPIL3(I) = YPIL3(I) - YREFER                                      VCR07260
      ZPIL3(I) = ZPIL3(I) - ZREFER                                      VCR07270
      XC = XPIL3(I)ACOS(STANGR) + YPIL3(I)ASIN(STANGR)                  VCR07280
      YC = -XPIL3(I)ASIN(STANGR) + YPIL3(I)ACOS(STANGR)                 VCR07290
      XPIL3(I) = XC                                                     VCR07300
      YPIL3(I) = YC                                                     VCR07310
      WRITE(11,470) XPIL3(I), YPIL3(I), ZPIL3(I)                        VCR07320
  480 CONTINUE                                                          VCR07330
      WRITE(7,482)                                                      VCR07340
  482 FORMAT ('  ENTER MAXIMUM ACCEPTABLE LEVEL FOR PEAK PARTICLE ',    VCR07350
     X 'VELOCITY './/'  IN THE PILLAR')                                 VCR07360
      READ (7,A) PPVPAL                                                 VCR07370
      WRITE(11,484) PPVPAL                                              VCR07380
  484 FORMAT(I5)                                                        VCR07390
      GO TO 500                                                         VCR07400
C-----------------------------------------------------------------------VCR07410
C    AA        READ FROM FILE        AA                                 VCR07420
C-----------------------------------------------------------------------VCR07430
  490 IF (KODE.NE.'P') GO TO 500                                        VCR07440
      READ (11,280) NCHK                                                VCR07450
      DO 420 I = 1, NCHK                                                VCR07460
      READ (11,310) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)             VCR07470
  420 CONTINUE                                                          VCR07480
      READ(11,450) NUMPIL                                               VCR07490
      DO 498 I = 1, NUMPIL                                              VCR07500
      READ(11,470) XPIL1(I),YPIL1(I),ZPIL1(I)                           VCR07510
      READ(11,470) XPIL2(I),YPIL2(I),ZPIL2(I)                           VCR07520
      READ(11,470) XPIL3(I),YPIL3(I),ZPIL3(I)                           VCR07530
  498 CONTINUE                                                          VCR07540
      READ(11,484) PPVAL                                                VCR07550
  500 CONTINUE                                                          VCR07560
C-----------------------------------------------------------------------VCR07570
C   AA       ENTER DATA FOR THE EXPLOSIVE WHICH IS USED        AA       VCR07580
C-----------------------------------------------------------------------VCR07590
      WRITE(7,509)                                                      VCR07600
  509 FORMAT('  ENTER TYPE OF EXPLOSIVE TO BE USED :')                  VCR07610
      READ(7,1509) EXPTYP                                               VCR07620
 1509 FORMAT(A15)                                                       VCR07630
      WRITE(7,507)                                                      VCR07640
  507 FORMAT('  WHAT IS THE WEIGHT STRENGTH OF THE  EXPLOSIVE YOU ARE', VCR07650
     X' USING  ?')                                                      VCR07660
      READ (7,A) WSTR                                                   VCR07670
C-----------------------------------------------------------------------VCR07680
      WRITE(7,508)                                                      VCR07690
  508 FORMAT('  WHAT IS THE DENSITY OF THE EXPLOSIVE YOU ARE USING ?',/ VCR07700
```

```
     X           '      (GM/CC IN METRIC, LBS/CUBIC INCH IN IMPERIAL)')
      READ (7,A) RHOE
      WRITE (7,520)
  520 FORMAT(' ENTER STOPE LENGTH, STOPE WIDTH AND STOPE DEPTH :')
      READ (7,A) XMAX, YMAX, ZMAX
      XMIN = 0.0
      YMIN = 0.0
      ZMIN = 0.0
      RETURN
      END
      SUBROUTINE BLMOD(MAXP)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C   PURPOSE :                                                          C
C              TO DESIGN THE INITIAL BLOCK MODEL                       C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C    AAA   INPUT  AAA
C   XMAX : MAX DIMENSION OF THE BLAST IN THE X DIRECTION
C   XMIN : MIN   "      "  "   "    "   "  "   "
C   YMAX : MAX   "      "  "   "    "   "  Y   "
C   YMIN : MIN   "      "  "   "    "   "  "   "
C   ZMAX : MAX   "      "  "   "    "   "  Z   "
C   ZMIN : MIN   "      "  "   "    "   "  "   "
C   MAXP : MAXIMUM PERIOD (IPER(NUMBH))
C
C    AAA  OUTPUT  AAA
C   DX : DISTANCE BETWEEN TWO CONSEQUTIVE POINTS IN THE X DIRECTION
C   DY :    "        "     "       "        "    "  "  Y    "
C   DZ :    "        "     "       "        "    "  "  Z    "
C   IBLOCK (I,J,K) = 0
       COMMON /KDRDT1/ OCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),BHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XYMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE
       COMMON/IBLK/ IBLOCK(100,100,5),ITIME(100,100,5)
       COMMON/XBLK/ XB(100),YB(100),ZB(5)
       COMMON/DXDYDZ/ DX,DY,DZ
       COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT
      X ,STDEPT
       IBIG = 3000
       DX = (XMAX - XMIN)/100.0
       DY = (YMAX - YMIN)/100.0
       DZ = (ZMAX - ZMIN)/5.0
       DO 10 I=1,100
       DO  9 J=1,100
       DO  8 K=1,5
        IBLOCK(I,J,K) = 0
        ITIME(I,J,K) = IBIG
    7 CONTINUE
       ZB(K) = ZMAX - DZAFLOAT(K)
    8 CONTINUE
```

```
      YB(J) = YMIN + DY*FLOAT(J-1)
    9 CONTINUE
      XB(I) = XMIN + DX*FLOAT(I-1)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CRPLT1(IA.L)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C   PURPOSE  :                                                         C
C            TO DESIGN CRATERS WITH VERTICAL AXES                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  :
C       **   INPUT   **
C      IA : NUMBER OF HOLE
C      X(IA) : X COORDINATE OF IA HOLE
C      Y(IA) : Y     "       "  "   "
C      Z(IA) : Z     "       "  "   "
C      XB(I) : X     "       "  I  POINT  OF THE BLOCK MODEL
C      YB(I) : Y     "       "  I    "     "  "    "     "
C      ZB(I) : Z     "       "  I    "     "  "    "     "
C
C       **  OUTPUT  **
C      IBLOCK(I,J,K) = K IF POINT (I,J,K) IS TAKEN
C
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC.XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10).YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE
      COMMON/IBLK/ IBLOCK(100,100.5), ITIME(100,100.5)
      COMMON/ISEQ/ IPER(50),KK
      COMMON/CHARGE/ WEIGHT(50),DOPT(50),RCR(50),DEPTH(50),SCRUP
      COMMON/XBLK/ XB(100),YB(100),ZB(5)
      COMMON/DXDYDZ/ DX,DY,DZ
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT
     X ,STDEPT
      RNEW =(RCR(IA)+RCR(IA)*(DEPTH(IA)-DZ)/DEPTH(IA))/2.0
C-----------------------------------------------------------------------
C    **  DESIGN THE CRATER  **
C-----------------------------------------------------------------------
      DO 20 K = 1, 5
      DO 15 I = 1. 100
      DO 10 J = 1. 100
      CALL DISTPT( X(IA), Y(IA), Z(IA), XB(I), YB(J), Z(IA), D)
      IF (D.GT.RNEW) GO TO 10
      IBLOCK(I,J,K) = K
      ITIME(I,J,K) = L
   10 CONTINUE
   15 CONTINUE
      ANOM = (DEPTH(IA)-DZ*FLOAT(K))/DEPTH(IA)
      ANOM1 = (DEPTH(IA)-DZ*FLOAT(K+1))/DEPTH(IA)
      IF(ANOM1.GT.0.0) GO TO 68
```

```
      ANOM1=0.0                                                          VCR088
      IF(ANOM.GT.0.0) GO TO 68                                           VCR088
      RNEW = 0.0                                                         VCR088
      GO TO 20                                                           VCR088
   68 RNEW =  RCR(IA) * (ANOM + ANOM1)/2.0                               VCR088
   20 CONTINUE                                                           VCR088
      RETURN                                                             VCR088
      END                                                                VCR088
      SUBROUTINE CRPLT2 (ICR)                                            VCR088
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR089
C                                                                      C VCR089
C   PURPOSE  :                                                         C VCR089
C            TO CONSIDER THE EFFECT OF THE EXISTING CRATERS IN THE     C VCR089
C            DETONATION OF A NEW CHARGE .                              C VCR089
C                                                                      C VCR089
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR089
C                                                                        VCR089
C   VARIABLES  :                                                         VCR089
C                                                                        VCR089
C        **   INPUT   **                                                 VCR090
C     IPER(IFIRST): PREVIOUS PERIOD                                      VCR090
C     IBLOCK(II,JJ,KKK) = KKK- POINT (II,JJ,KK) IS TAKEN IN PERIOD L     VCR090
C     X(ICR).Y(ICR).Z(ICR) : X Y Z COORDINATES  OF THE CENTRE OF THE     VCR090
C     CHARGE ICR                                                         VCR090
C     XB(II),YB(JJ),ZB(KKK) : X Y Z COORDINATES OF POINT (BLOCK)         VCR090
C     (II,JJ,KKK)                                                        VCR090
C     NUMBH : NUMBER OF HOLES                                            VCR090
C     BURDEN(ICR,IA) : BURDEN OF CHARGE ICR TO CRATER IA.                VCR090
C     DCONCH : DEPTH OF CONTAINED CHARGE********                         VCR090
C     DELTAR(MI) : DEPTH RATIO FOR THE SCALED RADIUS CURVE (POINT MI)    VCR091
C     WEIGHT(ICR) : WEIGHT OF CHARGE ICR                                 VCR091
C     DOPT(ICR) : DEPTH OF BURIAL OF CHARGE ICR                          VCR091
C        **   OUTPUT   **                                                VCR091
C     XS.YS.ZS : ARRAYS DEFINING A CLOSED POLYGON                        VCR091
C     IBLOCK(I,J,K) = 0 IF BLOCK (I,J,K) IS TO BE TAKEN.                 VCR091
C----------------------------------------------------------------------- VCR091
      COMMON/IBLK/ IBLOCK(100.100.5), ITIME(100.100.5)                   VCR091
      COMMON/XBLK/ XB(100),YB(100),ZB(5)                                 VCR091
      COMMON/DXDYDZ/ DX,DY,DZ                                            VCR091
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH.SCDEPT(50),DELTA(50),         VCR092
     XSCRAD(50).DHOLE.X(50),Y(50),Z(50),XVCHK(10).YVCHK(10).ZVCHK(10)    VCR092
     X,PPVALL(10),AKFAC.AMFAC,XPIL1(10),XPIL2(10),XPIL3(10).YPIL1(10),   VCR092
     XYPIL2(10),YPIL3(10).ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,     VCR092
     XXMAX,XMIN.YMAX,YMIN,ZMAX,ZMIN,DELTAR(30),RHOE                      VCR092
      COMMON/RDRDT2/ NP1,NP2.NUMBH,NCHK.NUMPIL,INIT(50).MSYSUN           VCR092
      COMMON/DELT/ DELTO. EFACTR                                         VCR092
      COMMON/ISEQ/ IPER(50).KK                                           VCR092
      COMMON/CHARGE/ WEIGHT(50),DOPT(50),RCR(50).DEPTH(50),SCROP         VCR092
      COMMON/BURDEN/BURDEN(50,50)                                        VCR092
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR.STLENG,STWIDT            VCR092
     X .STDEPT                                                           VCR092
      DIMENSION BURMIN(50),XS(10),YS(10)                                 VCR092
      BIG = 10000.0                                                      VCR092
      L = IPER(ICR) - 1                                                  VCR092
C----------------------------------------------------------------------- VCR092
```

```
C  **  FIND CRATERS AT PERIOD L  **                                    VCR09360
C-----------------------------------------------------------------------VCR09370
      DO 70 NI = 1, NUMBH                                               VCR09380
      IF (IPER(NI).GT.L) GO TO 60                                       VCR09390
      BURMIN(NI) = BURDEN(ICR,NI)                                       VCR09400
      GO TO 70                                                          VCR09410
   60 BURMIN(NI) = BIG                                                  VCR09420
   70 CONTINUE                                                          VCR09430
C-----------------------------------------------------------------------VCR09440
C  **      FIND MINIMUM BURDEN       **                                 VCR09450
C-----------------------------------------------------------------------VCR09460
      BMIN = BURMIN(1)                                                  VCR09470
      DO 80 NI = 2, NUMBH                                               VCR09480
      IF (BMIN.GT.BURMIN(NI)) BMIN = BURMIN(NI)                         VCR09490
   80 CONTINUE                                                          VCR09500
C-----------------------------------------------------------------------VCR09510
C  ** IF BMIN GREATER THAN DEPTH OF CONTAINED CHARGE, POINT DOES NOT    VCR09520
C          AFFECT DETONATION                                            VCR09530
C-----------------------------------------------------------------------VCR09540
      DCONT = EFACTR * WEIGHT(ICR)**(1.0/3.0)                           VCR09550
      IF (BMIN.GT.DCONT) GO TO 2222                                     VCR09560
C-----------------------------------------------------------------------VCR09570
C  **   FIND CRATER RADIUS FOR MINIMUM BURDEN   **                      VCR09580
C-----------------------------------------------------------------------VCR09590
      DEPR = BMIN / DCONT                                               VCR09600
C-----------------------------------------------------------------------VCR09610
C  **  FIND SCALED RADIUS AT THIS DEPTH RATIO  **                       VCR09620
C-----------------------------------------------------------------------VCR09630
      DO 40  MI = 2, NP2                                                VCR09640
      IF ((DELTAR(MI).GE.DEPR).AND.(DELTAR(MI-1).LE.DEPR)) GO TO 35     VCR09650
   40 CONTINUE                                                          VCR09660
   35 SCR = SCRAD(MI-1) + (DEPR - DELTA(MI-1))/(DELTA(MI)               VCR09670
     XDELTAR(MI-1))*(SCRAD(MI)-SCRAD(MI-1))                             VCR09680
      RADIUS = SCR * WEIGHT(ICR)**(1.0/3.0)                             VCR09690
C-----------------------------------------------------------------------VCR09700
C  **   CALCULATE SLANT DISTANCE   **                                   VCR09710
C-----------------------------------------------------------------------VCR09720
      SLANTD = (RADIUS**2.0 + BMIN**2.0)**0.5                           VCR09730
      DO 2020 II = 2, 99                                                VCR09740
      DO 2010 JJ = 2, 99                                                VCR09750
      DO 2000 KKK = 1, 5                                                VCR09760
C-----------------------------------------------------------------------VCR09770
C  **  CHECK IF POINT (II,JJ,KKK) IN PERIOD L IS INSIDE THE MASS        VCR09780
C       AND THEREFORE IT DOES NOT  AFFECT THE DETONATION     **         VCR09790
C-----------------------------------------------------------------------VCR09800
      IF(ITIME(II,JJ,KKK).GT.L)GO TO 2000                               VCR09810
      III=II+1                                                          VCR09820
      JJ1=JJ+1                                                          VCR09830
      IIM=II-1                                                          VCR09840
      JJM=JJ-1                                                          VCR09850
      IF((IBLOCK(III,JJ,KKK).NE.0).AND.(IBLOCK(II,JJ1,KKK).NE.0).AND.   VCR09860
     X(IBLOCK(IIM,JJ,KKK).NE.0).AND.(IBLOCK(II,JJM,KKK).NE.0))          VCR09870
     XGO TO 2000                                                        VCR09880
C-----------------------------------------------------------------------VCR09890
C  ** POINT MAY AFFECT DETONATION. CALCULATE DISTANCE BETWEEN POINT     VCR09900
```

```
C     AND THE CENTRE OF THE CHARGE.
C-----------------------------------------------------------------------
      ZCHARG = Z(ICR)-DOPT(ICR)
      CALL DISTPT(X(ICR),Y(ICR),ZCHARG,XB(II),YB(JJ),ZB(KKK),
     X DIST)
C-----------------------------------------------------------------------
C ** BLOCKS FURTHER THAN THE SLANT DISTANCE DO NOT AFFECT DETONATION **
C-----------------------------------------------------------------------
      IF(DIST.GT.SLANTD) GO TO 2010
      DO 1025 I1 = 1,100
      DO 1020 J1 = 1,100
      KK1 = 1
      IF((XB(I1).GT.X(ICR)).AND.(XB(I1).GT.(XB(I1)+DX))) GO TO 1020
      IF((XB(I1).GT.X(ICR)).AND.(XB(I1).LT.(X(ICR)-DX))) GO TO 1020
      IF((XB(I1).LT.X(ICR)).AND.(XB(I1).LT.(XB(I1)-DX))) GO TO 1020
      IF((XB(I1).LT.X(ICR)).AND.(XB(I1).GT.(X(ICR)+DX))) GO TO 1020
      IF((YB(JJ).GT.Y(ICR)).AND.(YB(J1).GT.(YB(JJ)+DY))) GO TO 1020
      IF((YB(JJ).GT.Y(ICR)).AND.(YB(J1).LT.(Y(ICR)-DY))) GO TO 1020
      IF((YB(JJ).LT.Y(ICR)).AND.(YB(J1).LT.(YB(JJ)-DY))) GO TO 1020
      IF((YB(JJ).LT.Y(ICR)).AND.(YB(J1).GT.(Y(ICR)+DY))) GO TO 1020
      CALL DISPLN(XB(I1),YB(JJ),X(ICR),
     X Y(ICR),XB(I1),YB(J1),DIS,IBUG)
       IF(IBUG.EQ.1) GO TO 2010
C-----------------------------------------------------------------------
C   **    TAKE ZONE OF THICKNESS DX      **
C-----------------------------------------------------------------------
      IF (DIS.GT.(DX/2.)) GO TO 1020
      DO 1000 KK1 = 1,5
      IF(IBLOCK(I1,J1,KK1).NE.0) GO TO 1000
C-----------------------------------------------------------------------
C   **    TRANSLATE COORDINATES     **
C-----------------------------------------------------------------------
      XS(1) = 0.0
      YS(1) = Z(ICR) - DOPT(ICR)
      XS(2) = 0.0
      YS(2) = Z(ICR)
      XS(3) = (ABS(X(ICR)-XB(I1))**2.0+
     X ABS(Y(ICR)-YB(JJ))**2.0)**0.5
      AAAA=-XS(3)
      IF(XB(I1).LT.X(ICR)) XS(3)=AAAA
      IF((XB(I1).EQ.X(ICR)).AND.(YB(JJ).LT.Y(ICR))) XS(3)=AAAA
      YS(3) = Z(ICR)
      XS(4) = XS(3)
      YS(4) = ZB(KKK)
      XS0=(ABS(X(ICR)-XB(I1))**2.0+ABS(Y(ICR)-YB(J1))**2.0)**0.5
      AAAA0=-XS0
      IF(XB(I1).LT.X(ICR)) XS0 = AAAA0
      IF((XB(I1).EQ.X(ICR)).AND.(YB(J1).LT.YB(ICR))) XS0 = AAAA0
      YS0 = ZB(KK1)
      CALL INSIDE(4,XS,YS,XS0,YS0,KKODE)
      IF(KKODE.EQ.0) GO TO 1000
      IBLOCK(I1,J1,KK1) = KK1
      ITIME(I1,J1,KK1) = IPER(ICR)
 1000 CONTINUE
 1020 CONTINUE
```

```
 1025 CONTINUE
 2000 CONTINUE
 2010 CONTINUE
 2020 CONTINUE
 2222 RETURN
      END
      SUBROUTINE INSIDE (NPN,X,Y,XO,YO,KODE)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C   PURPOSE :                                                         C
C           TO DETERMINE IF A POINT LIES WITHIN A CLOSED              C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     BOUNDARY
C     NPN : THE NUMBER OF POINTS DEFINING THE CLOSED POLYGON
C     X,Y : THE ARRAYS WITHE THE ABOVE POINT COORDINATES
C     KODE: =-1 THE POINT IS INSIDE BOUNDARY
C     KODE: =0  THE POINT IS OUTSIDE BOUNDARY
C     KODE: =I  THE POINT IS ON THE INSIDE BOUNDARY  IN VECTOR (I-1,I)
C----------------------------------------------------------------------
      DIMENSION X(10),Y(10)
C
C  MINIMUM POINTS 3  MAXIMUM POINTS 999
C----------------------------------------------------------------------
      IF (NPN.LE.2.OR.NPN.GT.1999)  RETURN
      PP=3.141592
      KODE=-2
C
       X1=X(NPN)-XO
       Y1=Y(NPN)-YO
       SUM=0.0
       NTELOS=NPN
       DO 401 KPN=1,NTELOS
       ANGL=0.0
       X2=X(KPN)-XO
       Y2=Y(KPN)-YO
       IF (X2.EQ.X1.AND.Y2.EQ.Y1) GO TO 401
       ANOM=X1*Y2-X2*Y1
       DENOM=X1*X2+Y1*Y2
       IF (DENOM.NE.0.0) ANGL=ATAN(ANOM/DENOM)
       IF (DENOM.EQ.0.0) ANGL=PP/2.
       IF (DENOM.EQ.0.0.AND.ANOM.LT.0.0) ANGL=-PP/2.
       X1=X2
       Y1=Y2
       IF (DENOM.LT.0.0.AND.ANGL.EQ.0.0) KODE=KPN
       IF (KODE.EQ.KPN) RETURN
C---------------------------------------------------------------------
C   IF ANGL=180. THE POINT FALLS ON THE AREA BOUNDARY
C---------------------------------------------------------------------
      IF (ANGL.LT.0.0.AND.ANOM.GT.0.0) ANGL=ANGL+PP
      IF (ANGL.GT.0.0.AND.ANOM.LT.0.0) ANGL=ANGL-PP
      SUM=SUM+ANGL
  401 CONTINUE
      P22 = 2.0* PP
      P44 = 4.0* PP
```

```
      P66 = 6.0* PP
C-----------------------------------------------------------------------
C    IF SUM =0. THE POINT FALLS OUTSIDE THE AREA BOUNDARY
C-----------------------------------------------------------------------
      IF (ABS(SUM).LT.0.3) KODE=0
      IF (ABS(SUM).LT.0.3) RETURN
C-----------------------------------------------------------------------
C    IF SUM =2*PP THE POINT FALLS INSIDE THE AREA BOUNDARY
C-----------------------------------------------------------------------
      IF (ABS(SUM).GT.(2.0*PP-0.3)) KODE=-1
      IF (KODE.EQ.(-1)) RETURN
C
C   SHOULD NOT GET HERE. ENTER AN ERROR CODE
C-----------------------------------------------------------
      KODE = -2
      RETURN
      END
      SUBROUTINE DISTPL(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X,Y,Z,D,IBUG)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PURPOSE :                                                           C
C            TO CALCULATE THE DISTANCE FROM A POINT TO A PLANE         C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES  :
C  ***  INPUT  ***
C    X1,Y1,Z1 : THE X, Y, Z COORDINATES OF ONE POINT OF THE PLANE
C    X2,Y2,Z2 : THE X, Y, Z COORDINATES OF A SECOND POINT OF THE PLANE
C    X3,Y3,Z3 : THE X, Y, Z COORDINATES OF A THIRD POINT OF THE PLANE
C    X, Y, Z  : THE POINT OUTSIDE OF THE PLANE
C  ***  OUTPUT  ***
C    D        : THE DISTANCE  OF THE POINT FROM THE PLANE
      IBUG = 0
      A1 = (X2-X1)*(Z3-Z1) - (Z2-Z1)*(Y3-Y1)
      B1 = (Z2-Z1)*(X3-X1) - (X2-X1)*(Z3-Z1)
      C1 = (X2-X1)*(Y3-Y1) - (Y2-Y1)*(X3-X1)
      D1 = Z1*(Y2-Y1)*(X3-X1) + X1*(Z2-Z1)*(Y3-Y1) +
     X     Y1*(X2-X1)*(Z3-Z1) - X1*(Y2-Y1)*(Z3-Z1) -
     X     Y1*(Z2-Z1)*(X3-X1) - Z1*(X2-X1)*(Y3-Y1)
      A2 = ABS(A1)
      B2 = ABS(B1)
      C2 = ABS(C1)
      DENOM = A2**2.0 + B2**2.0 + C2**2.0
      IF(DENOM.LE.1E-15) GO TO 10
      DD= (A1*X + B1*Y + C1*Z + D1) / (A2**2.0 + B2**2.0 + C2**2.0)**0.5
      D = ABS (DD)
      RETURN
   10 IBUG = 1
      RETURN
      END
      SUBROUTINE DISTPT(X1,Y1,Z1,X,Y,Z,D)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PURPOSE :                                                           C
C            TO CALCULATE THE DISTANCE BETWEEN TWO POINTS.             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C
C  VARIABLES  :
C
C    ***  INPUT  ***
C   X,Y,Z,X1,Y1,Z1 : THE X, Y, Z COORDINATES OF THE TWO POINTS.
C
C    *** OUTPUT  ***
C   D : THE DISTANCE
C
      A = ABS(X-X1)
      B = ABS(Y-Y1)
      C = ABS(Z-Z1)
      D = (A**2.0 + B**2.0 + C**2.0)**0.5
      RETURN
      END
      SUBROUTINE VBRCHK (IFIRST)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C  PURPOSE :                                                            C
C             TO EXAMINE THE VIBRATIONS RESULTING FROM THE DETONATIONS  C
C             OF SINGLE AND MULTIPLE CHARGES.                           C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  :
C
C       **    INPUT    **
C   NUMBH  : NUMBER OF CHARGED HOLES
C   IPER(IFIRST) : PRESENT DELAY PERIOD
C   DHOLE :  HOLE DIAMETER
C   RHOE : DENSITY OF THE EXPLOSIVE
C   WSTR : WEIGHT STRENGTH OF EXPLOSIVE USED (RELATIVE TO ANFO)
C   WSTRTH : WEIGHT STRENGTH OF EXPLOSIVE AT TESTS TO OBTAIN CURVES.
C   XCHK,YCHK,ZCHK : ARRAYS OF X Y Z COORDINATES OF SPECIFIC POINTS
C   FOR VIBRATION CHECKS.
C   X,Y,Z : ARRAYS FOR THE COORDINATES OF THE HOLES
C   NCHK : NUMBER OF POINTS TO BE CHECKED FOR VIBRATIONS
C   AMFAC, AKFAC : SITE FACTORS FOR THE VIBRATIONS
C   PPVALL(I) : MAX ACCEPTABLE PPV IN POINT I
C   NUMPIL : NUMBER OF PILLARS
C   XPIL1(I),YPIL1(I),ZPIL1(I),.... COORDINATES OF THE POINTS DEFINING
C   PILLAR I
C   PPVPAL(I) : MAX ACCEPTABLE PPV IN PILLARS
C   DELTO : OPTIMUM DEPTH RATIO
C   EFACTR : LIVINGSTON'S STRAIN ENERGY FACTOR
C   SCRAD(IOPT) : SCALED RADIUS OF POINT IOPT
C   IFLY(I) : I IN THE FLYROCK REGION
C
C     **  OUTPUT   **
C
C   WEIGHT(IA) : WEIGHT OF CHARGE IA
C   DOPT(IA) : OPTIMUM DEPTH OF BURIAL
C   RCR(IA) : CRATER RADIUS
C
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
```

```
      XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)    VCR12110
      X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),   VCR12120
      XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,     VCR12130
      XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE                      VCR12140
       COMMON/RDRDT2/ NP1,NP2,NUMBH,NCHK,NUMPIL,INIT(50),MSYSUN           VCR12150
       COMMON/CHARGE/WEIGHT(50),DOPT(50),RCR(50),DEPTH(50),SCROP          VCR12160
       COMMON/FLY/IFLY(50)                                                VCR12170
       COMMON/DELT/DELTO,EFACTR                                           VCR12180
       COMMON/ISEQ/ IPER(50),KK                                           VCR12190
       COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT            VCR12200
      X .STDEPT                                                           VCR12210
       DIMENSION ALOAD(50),PPV(10),PPVPIL(10),WEIGMX(50),ID(20)           VCR12220
       WEIGMX(1) = 100000.0                                               VCR12230
       ICH = 0                                                            VCR12240
       DO 2 I = 1, 20                                                     VCR12250
       ID(I)=0                                                            VCR12260
     2 CONTINUE                                                           VCR12270
C-----------------------------------------------------------------------  VCR12280
C  ** FIND CHARGES DETONATING IN THE SAME PERIOD  **                      VCR12290
C-----------------------------------------------------------------------  VCR12300
       KK = 0                                                             VCR12310
       DO 100  IA = 1, NUMBH                                              VCR12320
       IF (IPER(IA).NE.IPER(IFIRST)) GO TO 100                            VCR12330
C-----------------------------------------------------------------------  VCR12340
C  ** LOAD HOLE (LENGTH / DIAMETER = 6/1)  **                             VCR12350
C-----------------------------------------------------------------------  VCR12360
       PI = 3.14159                                                       VCR12370
       ALOAD(IA) = (6 * DHOLE * PI * DHOLE ** 2.0 / 4.0) * RHOE           VCR12380
       IF(MSYSUN.EQ.1) ALOAD(I) = ALOAD(I)/1000.0                         VCR12390
C-----------------------------------------------------------------------  VCR12400
C  ** CALCULATE EQUIVALENT WEIGHT OF EXPLOSIVE  **                        VCR12410
C-----------------------------------------------------------------------  VCR12420
       WEIGHT(IA) = ALOAD(IA) * WSTR / WSTRTH                             VCR12430
C-----------------------------------------------------------------------  VCR12440
C  **   CHECK VIBRATION LEVELS   **                                       VCR12450
C-----------------------------------------------------------------------  VCR12460
       IF(NCHK.EQ.0) GO TO 32                                             VCR12470
C-----------------------------------------------------------------------  VCR12480
C  ** CALCULATE DISTANCES FROM POINTS                                     VCR12490
C-----------------------------------------------------------------------  VCR12500
       DO 30 I = 1, NCHK                                                  VCR12510
       CALL DISPT(XVCHK(I),YVCHK(I),ZVCHK(I),X(IA),Y(IA),Z(IA),DIST)      VCR12520
C-----------------------------------------------------------------------  VCR12530
C  ** CALCULATE PPV AT THIS POINT  **                                     VCR12540
C-----------------------------------------------------------------------  VCR12550
       PPV(I) = AKFACA(DIST/WEIGHT(IA)**(1.0/3.0))**AMFAC                 VCR12560
C-----------------------------------------------------------------------  VCR12570
C  ** FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT **                    VCR12580
C-----------------------------------------------------------------------  VCR12590
       WEIGMX(I) = (AKFAC/PPVALL(I))**(3.0/AMFAC)*DIST**3.0               VCR12600
    30 CONTINUE                                                           VCR12610
C-----------------------------------------------------------------------  VCR12620
C  ** CALCULATE DISTANCE FROM PILLARS  **                                 VCR12630
C-----------------------------------------------------------------------  VCR12640
    32 IF (NUMPIL.EQ.0) GO TO 52                                          VCR12650
```

```
      DO 50 I = 1, NUMPIL                                               VCR12660
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL3(I),YPIL2(I),ZPIL2(I),VCR12670
     XXPIL3(I),YPIL3(I),ZPIL3(I),X(IA),Y(IA),Z(IA),DISTP)               VCR12680
C-----------------------------------------------------------------------VCR12690
C  ** CALCULATE PPV AT THIS POINT **                                    VCR12700
C-----------------------------------------------------------------------VCR12710
      PPVPIL(I) = AKFAC * (DISTP/WEIGHT(IA)**(1.0/3.0))**               VCR12720
     X AMFAC                                                            VCR12730
C-----------------------------------------------------------------------VCR12740
C  ** FIND MAXIMUM WEIGHT BECAUSE OF THIS PARTICULAR PILLAR **          VCR12750
C-----------------------------------------------------------------------VCR12760
      WEIGMX(NCHK+1) = (AKFAC/PPVPAL)**(3.0/AMFAC)*DISTP**3.0           VCR12770
   50 CONTINUE                                                          VCR12780
C-----------------------------------------------------------------------VCR12790
C  **  FIND MINIMUM ACCEPTABLE WEIGHTS  **                              VCR12800
C-----------------------------------------------------------------------VCR12810
   52 WMIN = 2000000.0                                                  VCR12820
      NN = NCHK + NUMPIL                                                VCR12830
      IF (NN.EQ.0) GO TO  99                                            VCR12840
      DO 80 I = 1, NN                                                   VCR12850
      IF (WEIGMX(I).LT.WMIN) WMIN=WEIGMX(I)                             VCR12860
   80 CONTINUE                                                          VCR12870
      IF (WEIGHT(IA).LE.WMIN) GO TO 99                                  VCR12880
      WEIGHT(IA) = WMIN                                                 VCR12890
   99 KK = KK + 1                                                       VCR12900
      INIT(KK) = IA                                                     VCR12910
C-----------------------------------------------------------------------VCR12920
C  ** CALCULATE OPTIMUM DEPTH OF BURIAL **                              VCR12930
C-----------------------------------------------------------------------VCR12940
      DOPT(IA) = DELTO * CFACTR * (WEIGHT(IA))**(1.0/3.0)               VCR12950
      DEPTH(IA) = DOPT(IA)                                              VCR12960
C-----------------------------------------------------------------------VCR12970
C  ** CALCULATE CRATER RADIUS **                                        VCR12980
C-----------------------------------------------------------------------VCR12990
      RCR(IA) = SCROP * WEIGHT(IA) ** (1.0/3.0)                         VCR13000
  100 CONTINUE                                                          VCR13010
      IF(NN.EQ.0) GO TO 250                                             VCR13020
C-----------------------------------------------------------------------VCR13030
C       CHECK VIBRATIONS DUE TO THE SIMULTANEOUS  DETONATION OF         VCR13040
C                  MANY CHARGES                                         VCR13050
C-----------------------------------------------------------------------VCR13060
C                                                                       VCR13070
C-----------------------------------------------------------------------VCR13080
C  ** FIND AN AVERAGE POSITION OF THE CONCENTRATED CHARGE **            VCR13090
C-----------------------------------------------------------------------VCR13100
  105 XM = 0.0                                                          VCR13110
      IF(KK.EQ.1) GO TO 200                                             VCR13120
      YM = 0.0                                                          VCR13130
      ZM = 0.0                                                          VCR13140
      WM = 0.0                                                          VCR13150
      DO 110  I = 1,KK                                                  VCR13160
      IM = INIT(I)                                                      VCR13170
      XM = XM + X(I)*WEIGHT(IM)                                         VCR13180
      YM = YM + Y(I)*WEIGHT(IM)                                         VCR13190
      ZM = ZM + Z(I)*WEIGHT(IM)                                         VCR13200
```

```
      WM = WM + WEIGHT(IM)
  110 CONTINUE
      XM = XM / WM
      YM = YM / WM
      ZM = ZM / WM
C-----------------------------------------------------------------------
C  **  CALCULATE DISTANCE FROM POINTS  **
C-----------------------------------------------------------------------
      IF (NCHK.EQ.0) GO TO 131
      DO 130 I = 1, NCHK
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),XM,YM,ZM,DISTM)
C-----------------------------------------------------------------------
C  **   CALCULATE PPV AT THIS POINT   **
C-----------------------------------------------------------------------
      PPV(I) = AKFACA(DISTM/WMAA(1.0/3.0)) AA AMFAC
C-----------------------------------------------------------------------
C  **  FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT  **
C-----------------------------------------------------------------------
      WEIGMX(I) = (AKFAC/PPVALL(I))AA(3.0/AMFAC)ADISTMAA3.0
  130 CONTINUE
C-----------------------------------------------------------------------
C  **       SAME FOR PILLARS
C-----------------------------------------------------------------------
  131 IF (NUMPIL.EQ.0) GO TO 151
      DO 150 I = 1, NUMPIL
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL2(I),YPIL2(I),ZPIL2(I),
     X XPIL3(I),YPIL3(I),ZPIL3(I),XM,YM,ZM,DISTM)
      PPVPIL(I) = AKFACA((DISTM)/WMAA(1.0/3.0))AAAMFAC
      WEIGMX(NCHK+I) = (AKFAC/PPVPAL)AA(3.0/AMFAC)ADISTMAA3.0
  150 CONTINUE
C-----------------------------------------------------------------------
C  **    FIND MINIMUM OF ACCEPTABLE WEIGHTS    **
C-----------------------------------------------------------------------
  151 WMIN = 2000000.0
      NN = NCHK + NUMPIL
      IF (NN.EQ.0) GO TO 200
      DO 180 I = 1,NN
      IF(WEIGMX(I).LT.WMIN) WMIN = WEIGMX(I)
  180 CONTINUE
      IF (WM.LE.WMIN) GO TO 200
      IM = INIT(KK)
      KK = KK - 1
      ICH = ICH + 1
      ID(ICH) = IM
      GO TO 105
  200 IF(ICH.LQ.0) GO TO 250
      DO 210 I = 1, NUMBH
      IF (IPER(I).LE.IFIRST) GO TO 210
      IPER(I) = IPER(I) + 1
  210 CONTINUE
      DO 220 I = 1, ICH
      IC = ID(I)
      IPER(IC) = IFIRST + 1
  220 CONTINUE
      DO 235 I = 1, NUMBH
```

```
      DO 230 II = 1, ICH
      ISK = ID(II)
      IF (IFLY(ISK).EQ.I) IPER(I) = IPER(ISK)
      IF (IFLY(I).EQ.ISK) IPER(ISK) = IPER(I)
      IF (KK.GT.1) GO TO 230
      IF(IFLY(ISK).EQ.I) WEIGHT(ISK) = WEIGHT(ISK)-(WMIN-WM)
      IF(IFLY(I).EQ.ISK) WEIGHT(I) = WEIGHT(I)-(WMIN-WM)
  230 CONTINUE
  235 CONTINUE
  250 RETURN
      END
      SUBROUTINE BRDCHK(IFIRST,KIND,INEXT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C   PURPOSE  :                                                        C
C                 TO CHECK BURDENS OF NEW CHARGE TO THE PREVIOUS      C
C                  CRATERS AND DETERMINE IF THE CHARGE IS IN          C
C                  THE FLYROCK REGION.                                C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  :
C
C        **  INPUT  **
C      NUMBH : NUMBER OF HOLES
C      IPER(I) : PERIOD IN WHICH I DETONATES
C      IFIRST : CHARGE OF THE PREVIOUS PERIOD
C      KK : NUMBER OF CHARGES DETONATING IN PERIOD I
C      DEPTH(IA) : DEPTH OF CRATER IA
C      RCR(IA) : RADIUS OF CRATER IA ( MEASURED AT THE INITIAL FREE
C               FACE )
C      DOPT(INEXT) : DEPTH OF BURIAL OF CHARGE NO INEXT (OPTIMUM).
C        **  OUTPUT **
C      IFLY(I) : CHARGE RESULTING IN FLYROCK IF DETONATED AT THE INITIAL
C      PERIOD.
C      KIND : CODE NUMBER INDICATING FLYROCK IF EQUAL TO 2
C      BURDEN(INEXT,IA) : BURDEN FROM CHARGE INEXT TO CRATER IA
C      RATIO : RATIO (BURDEN TO ADJACENT CRATER / BURDEN TO HORIZONTAL
C      FREE FACE )
C----------------------------------------------------------------------
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE
      COMMON/ISEQ/ IPER(50), KK
      COMMON/CHARGE/WEIGHT(50),DOPT(50),RCR(50),DEPTH(50),SCROP
      COMMON/RDRDT2/ NP1,NP2,NUMBH,NCHK,NUMPIL,INIT(50),MSYSUN
      COMMON/BURDEN/ BURDEN(50,50)
      COMMON/FLY/IFLY(50)
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT
     X ,STDEPT
      DIMENSION  IPREV(50) , DIST(50,50)
C----------------------------------------------------------------------
C         ***    INITIALIZE    ***
```

```
C-----------------------------------------------------------------------
      DO 1190 IFF = 1 ,50
 1190 IFLY(IFF) = 0
C-----------------------------------------------------------------------
C     **      FIND PREVIOUS CRATERS        **
C-----------------------------------------------------------------------
      KPREV = 0
      DO 100 I = 1, NUMBH
      IF ( IPER(I).GE.IPER(IFIRST)) GO TO 100
      KPREV = KPREV + 1
      IPREV(KPREV) = I
  100 CONTINUE
C-----------------------------------------------------------------------
C     **  CHECK THE NEW CRATERS ONE BY ONE  **
C-----------------------------------------------------------------------
      DO 500 I = 1, KK
      INEXT = INIT(I)
      DO 400 II = 1, KPREV
      IA = IPREV (II)
      DIST(IA,INEXT) = (ABS(Y(INEXT) - Y(IA))**2.0
     X+ ABS(X(INEXT)-X(IA))**2.0)
     X**0.5
      A1 = DEPTH(IA)
      B1 = RCR(IA)
      C1 = -RCR(IA)*DEPTH(IA)
      BURDEN(INEXT,IA) = ABS((A1*DIST(IA,INEXT) + B1*DUPT(INEXT) +
     X C1 ) / (A1**2.0+B1**2.0) ** 0.5)
      RATIO = BURDEN(INEXT,IA) / DUPT(INEXT)
      IF (RATIO.LT.0.3) GO TO 1400
  400 CONTINUE
  500 CONTINUE
      KIND = 1
      RETURN
C-----------------------------------------------------------------------
C  **    FLYROCK REGION    **
C-----------------------------------------------------------------------
 1400 CONTINUE
C-----------------------------------------------------------------------
C  ** DETONATE INEXT AT THE SAME DELAY PERIOD WITH IA **
C-----------------------------------------------------------------------
      IPERT = IPER(INEXT)
      KTEST  = 0
      DO 1500 ITT = 1,NUMBH
      IF (IPER(ITT).EQ.IPERT) KTEST = KTEST + 1
 1500 CONTINUE
      IF (KTEST.EQ.1) GO TO 1510
      IPER(INEXT)=IPER(IA)
      GO TO 1521
 1510 DO 1520 ITT = 1,NUMBH
C     IF(IPER(ITT).GE.IPERT) IPER(ITT)=IPER(ITT)-(IPERT-
C    XIPER(IA))
      IF(IPER(ITT).GT.IPERT) IPER(ITT) = IPER(ITT) - 1
      IPER(INEXT) = IPER(IA)
 1520 CONTINUE
 1521 IFLY(INEXT) = IA
```

```
      KIND = 2                                                          VCR14860
      RETURN                                                            VCR14870
      END                                                               VCR14880
      SUBROUTINE DISPLN(X1,Y1,X2,Y2,X,Y,DISTA,IBUG)                     VCR14890
      IBUG = 0                                                          VCR14900
      A1= Y1 - Y2                                                       VCR14910
      B1= X2 - X1                                                       VCR14920
      C1= X1AY2-X2AY1                                                   VCR14930
      A = ABS(A1)                                                       VCR14940
      B = ABS(B1)                                                       VCR14950
      DENOM = AAA2.0 + BAA2.0                                           VCR14960
      IF(DENOM.LE.1.E-25) GO TO 10                                      VCR14970
      DISTA=ABS(A1AX + B1AY + C1)/DENOMAAO.5                            VCR14980
      RETURN                                                            VCR14990
   10 IBUG = 1                                                          VCR15000
      RETURN                                                            VCR15010
      END                                                               VCR15020
      SUBROUTINE SEQUEN                                                 VCR15030
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR15040
C                                                                       C VCR15050
C   PURPOSE  :                                                          C VCR15060
C              TO PROVIDE THE INITIATION SEQUENCE FOR AN UNDERGROUND    C VCR15070
C              BLAST USING THE VCR METHOD.                              C VCR15080
C                                                                       C VCR15090
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR15100
C                                                                       VCR15110
C   VARIABLES : REFER TO THE MAIN PROGRAM                               VCR15120
C                                                                       VCR15130
C                                                                       VCR15140
      CHARACTERA1 KKODE                                                 VCR15150
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),        VCR15160
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)   VCR15170
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),  VCR15180
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTK,    VCR15190
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE                     VCR15200
      COMMON/RDRDT2/ NP1,NP2,NUMBH,NCHK,NUMPIL,INIT(50),MSYSUN          VCR15210
      COMMON/IBLK/ IBLOCK(100,100,5),ITIME(100,100,5)                   VCR15220
      COMMON/DXDYDZ/ DX,DY,DZ                                           VCR15230
      COMMON/DELT/DELTU,LFACTK                                          VCR15240
      COMMON/XBLK/XB(100),YB(100),ZB(5)                                 VCR15250
      COMMON/ISEQ/ IPER(50),KK                                          VCR15260
      COMMON/FLY/ IFLY(50)                                              VCR15270
      COMMON/CHARGE/WEIGHT(50),DOPT(50),KCR(50),DEPTH(50),SCROP         VCR15280
      COMMON/BURDEN/ BURDEN(50,50)                                      VCR15290
      COMMON/TYPE/EXPTYP,EXPTYB                                         VCR15300
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT           VCR15310
     X ,STDEPT                                                          VCR15320
      DIMENSION ID(50),IEXAM(50),ADIST(50)                              VCR15330
C-----------------------------------------------------------------------VCR15340
C  AA      INITIALIZE DATA        AA                                    VCR15350
C-----------------------------------------------------------------------VCR15360
      DO 4 I=1,NUMBH                                                    VCR15370
      IPER(I) = 0                                                       VCR15380
      IEXAM(I) = 0                                                      VCR15390
    4 CONTINUE                                                          VCR15400
```

```
      IDELT = 1                                                         VCR154
      PI = 3.14159                                                      VCR1542
      RAD = SCRUPA((6ADHOLEAPIADHOLE**2.0/4.0)*RHOE*WSTR/WSTRTH)**      VCR1543
     X(1.0/3.0)                                                         VCR1544
      IF(MSYSUN.EQ.1)                                                   VCR1545
     XRAD=SCRUPA((6ADHOLEAPIADHOLE**2.0/4.0)*RHOE/1000.*WSTR/WSTRTH)** VCR1546
     X(1.0/3.0)                                                         VCR1547
      WRITE(7,10)                                                       VCR1548
   10 FORMAT('  ENTER (M) FOR MANUAL OR (A) FOR AUTOMATIC',/            VCR1549
     X        '          GENERATION OF THE INITIATION SEQUENCE.')       VCR1550
      READ (7,12) KKODE                                                 VCR1551
   12 FORMAT(1A1)                                                       VCR1552
      IF(KKODE.EQ.'A') GO TO 40                                         VCR1553
      WRITE(7,14)                                                       VCR1554
   14 FORMAT(/////,'              MANUAL GENERATION OF SEQUENCE',/       VCR1555
     X        '              ----------------------------',///)        VCR1556
      WRITE(7,20)                                                       VCR1557
   20 FORMAT('  ENTER NUMBER OF HOLE AND INITIATION PERIOD :')          VCR1558
      DO 25 I = 1,NUMBH                                                 VCR1559
      READ(7,A) NH,IPER(NH)                                             VCR1560
   25 CONTINUE                                                          VCR1561
      RETURN                                                            VCR1562
   40 WRITE(7,41)                                                       VCR1563
   41 FORMAT(/////,'              AUTOMATIC GENERATION OF SEQUENCE',/    VCR1564
     X        '              ---------------------------------',///)   VCR1565
      WRITE(7,43)                                                       VCR1566
   43 FORMAT('  ENTER THE NUMBER OF THE HOLE WHICH DETONATES FIRST :')  VCR1567
      READ (7,A) IF                                                     VCR1568
      IP = 1                                                            VCR1569
      IPER(IF) = 1                                                      VCR1570
   45 ICHCK1 = 0                                                        VCR1571
      DISTH = 100000.0                                                  VCR1572
      DO 46 I = 1,NUMBH                                                 VCR1573
      ADIST(I) = 1000.0                                                 VCR1574
   46 CONTINUE                                                          VCR1575
C-----------------------------------------------------------------------VCR1576
C  **     FIND NEXT HOLE     **                                         VCR1577
C-----------------------------------------------------------------------VCR1578
      DO 50 I = 1, NUMBH                                                VCR1579
      IF ( IPER(I).NE.0 ) GO TO 50                                      VCR1580
      CALL DISTPT(X(IF),Y(IF),Z(IF),X(I),Y(I),Z(I),DIST)                VCR1581
      IF(DIST.GT.DISTH) GO TO 50                                        VCR1582
      DISTH = DIST                                                      VCR1583
      IS = I                                                            VCR1584
   50 CONTINUE                                                          VCR1585
      IF (DISTH.GE.10000.0) GO TO 120                                   VCR1586
      IP = IP + 1                                                       VCR1587
      IPER(IS) = IP                                                     VCR1588
      II = 1                                                            VCR1589
      ICHECK = 0                                                        VCR1590
      DO 70 I = 1, NUMBH                                                VCR1591
      CALL DISTPT(X(IF),Y(IF),Z(IF),X(I),Y(I),Z(I),DIST)                VCR1592
      IF((DIST.GT.(DISTH+RAD)).OR.(DIST.LT.DISTH).OR.(I.EQ.IS))         VCR1593
     X GO TO 70                                                         VCR1594
      IEXAM(I) = 1                                                      VCR1595
```

```
      ICHECK = 1
   70 CONTINUE
      IF(ICHECK.EQ.0) GO TO 45
   72 ICHCK1 = 0
      DO 75 I = 1, NUMBH
      IF(IEXAM(I).NE.1) GO TO 75
      CALL DISTPT(X(IS),Y(IS),Z(IS),X(I),Y(I),Z(I),ADIST(I))
      ICHCK1 = 1
   75 CONTINUE
      IF(ICHCK1.EQ.1) GO TO 79
      IIMAX = 0
      DO 77 I=1,NUMBH
      IF(IIMAX.GE.IPER(I)) GO TO 77
      IIMAX = IPER(I)
      IP = IIMAX
   77 CONTINUE
      GO TO 45
C-----------------------------------------------------------------------
C  **      FIND MINIMUM DISTANCE      **
C-----------------------------------------------------------------------
   79 ADISTM = 100000.0
      DO 90 I = 1, NUMBH
      IF (IPER(I).GT.0) GO TO 90
      IF (IEXAM(I).EQ.0) GO TO 90
      IF (ADISTM.LT.ADIST(I)) GO TO 90
      ADISTM = ADIST(I)
      IS = I
   90 CONTINUE
      IF (IDELT.EQ.1) GO TO 95
      IPER(IS) = IP
      IDELT = 1
      GO TO 100
   95 IPER(IS) = IP + 1
      IDELT = 0
  100 IEXAM(IS) = 0
      GO TO 72
  120 RETURN
      END
      SUBROUTINE CHARGL(DDOUPD)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    PURPOSE :                                                         C
C          TO PROVIDE THE DATA FOR THE GRAPHICS PROGRAM                C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  : REFER TO THE MAIN PROGRAM
C
C
      CHARACTER*4 STOPID
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT
     X ,STDEPT
      DIMENSION ZKNOWN(50,20),ALENGT(50,20),WEIGHT(50,20),IPER(50,20),
     X XDI(50),YDI(50),ZDI(50),STOPID(40),DDOUPD(50)
      REWIND 51
```

```
      PP = 3.1415926                                                    VCR16510
   27 FORMAT(4F10.2)                                                    VCR16520
   19 FORMAT(20A4)                                                      VCR16530
   20 FORMAT(4F10.2)                                                    VCR16540
      REWIND 70                                                         VCR16550
      READ(70,910) NUMBH,NCHKGS                                         VCR16560
  910 FORMAT(15I5)                                                      VCR16570
      READ(70,900) DIAMT                                                VCR16580
      DO 1000 I = 1,NCHKGS                                              VCR16590
      READ(70,900) (ZKNOWN(J,I),J=1,NUMBH)                              VCR16600
  900 FORMAT(8F10.2)                                                    VCR16610
 1000 CONTINUE                                                          VCR16620
      DO 1050 I = 1,NCHKGS                                              VCR16630
      READ(70,900) (WEIGHT(J,I),J=1,NUMBH)                              VCR16640
 1050 CONTINUE                                                          VCR16650
      DO 1100 I = 1,NCHKGS                                              VCR16660
      READ(70,900) (ALENGT(J,I),J=1,NUMBH)                              VCR16670
 1100 CONTINUE                                                          VCR16680
      IEXP = 1                                                          VCR16690
      DO 1200 I = 1,NCHKGS                                              VCR16700
      READ(70,910) (IPEK(J,I),J=1,NUMBH)                                VCR16710
 1200 CONTINUE                                                          VCR16720
      REWIND 61                                                         VCR16730
      DO 3000 IHOSEQ = 1,NUMBH                                          VCR16740
      READ(61,100) IJUNK,N1,IHOROW,IHOCOL,DDDDD,DIPNOM                  VCR16750
  100 FORMAT(4I5,2F10.2)                                                VCR16760
      IF(IJUNK.EQ.99) GO TO 3008                                        VCR16770
      WRITE(62,21) NCHKGS,IHOROW,IHOCOL                                 VCR16780
   21 FORMAT(15I5)                                                      VCR16790
      READ(61,101) XCOLAK,YCOLAK,ZCOLAK                                 VCR16800
  101 FORMAT(3F10.2)                                                    VCR16810
      READ(61,101,END=3008) ((XDI(I),YDI(I),ZDI(I)),I=1,N1)             VCR16820
      DO 155 ICH = 1,NCHKGS                                             VCR16830
      X1 = XCOLAK                                                       VCR16840
      Y1 = YCOLAK                                                       VCR16850
      Z1 = ZCOLAK - ZKEYEK                                              VCR16860
      DI = 0                                                            VCR16870
      DO 150 I = 1,N1                                                   VCR16880
      Z2 = ZDI(I) - ZKEYEK                                              VCR16890
      X2 = XDI(I)                                                       VCR16900
      Y2 = YDI(I)                                                       VCR16910
      IF(Z2.LT.ZKNOWN(IHOSEQ,ICH)) GO TO 109                            VCR16920
      ZLOCAL = ZKNOWN(IHOSEQ,ICH)                                       VCR16930
      RATIO = (ZKNOWN(IHOSEQ,ICH)-Z1)/(Z2-Z1)                           VCR16940
      DX1X2 = RATIO*(X1-X2)                                             VCR16950
      DY1Y2 = RATIO*(Y1-Y2)                                             VCR16960
      DI=DI+(ABS(DX1X2)**2.0+ABS(DY1Y2)**2.0+ABS(Z1-ZKNOWN(IHOSEQ,ICH)) VCR16970
     X**2.0)**0.5                                                       VCR16980
      GO TO 200                                                         VCR16990
  109 ZLOCAL = Z2                                                       VCR17000
      DI=DI + (ABS(X1-X2)**2.0+ABS(Y1-Y2)**2.0+ABS(Z1-Z2)**2.0)**0.5    VCR17010
      X1 = X2                                                           VCR17020
      Y1 = Y2                                                           VCR17030
      Z1 = Z2                                                           VCR17040
  150 CONTINUE                                                          VCR17050
```

```
  200 DUP = DI - ALENGT(IHOSEQ,ICH)/2.0                                   VCR17060
      DDO = DI + ALENGT(IHOSEQ,ICH)/2.0                                   VCR17070
      DDOUPD(IHOSEQ) = DDO                                                VCR17080
      ZDO = ZCOLAR-ZREFER-(ZKNOWN(IHOSEQ,ICH)+ALENGT(IHOSEQ,ICH)/2.0)     VCR17090
      ZUP = ZCOLAR-ZREFER-(ZKNOWN(IHOSEQ,ICH)-ALENGT(IHOSEQ,ICH)/2.0)     VCR17100
      WRITE(62,260) ZDO,ZUP,DDO,DUP,WEIGHT(IHOSEQ,ICH),DIAMT,IEXPLT,IPERVCR17110
     X(IHOSEQ,ICH)                                                        VCR17120
  260 FORMAT(6F10.2,2I5)                                                  VCR17130
  155 CONTINUE                                                            VCR17140
 3000 CONTINUE                                                            VCR17150
 3008 CONTINUE                                                            VCR17160
      LAST = 99                                                           VCR17170
      WRITE(62,21) LAST,IHOROW,IHOCOL                                     VCR17180
      RETURN                                                              VCR17190
      END                                                                 VCR17200
      SUBROUTINE PLVIEW (PLELEV,XHOL,YHOL,ZHOL,NOHOLS)                    VCR17210
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR17220
C                                                                       C VCR17230
C    PURPOSE :                                                          C VCR17240
C        TO GIVE THE X Y COORDINATES OF THE HOLES AT A PARTICULAR       C VCR17250
C        ELEVATION                                                      C VCR17260
C                                                                       C VCR17270
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC VCR17280
C                                                                         VCR17290
C     VARIABLES                                                           VCR17300
C        PLELEV : PLAN ELEVATION                                          VCR17310
C        STOPID : STOPE IDENTIFICATION (UP TO 80 CHARACTERS)              VCR17320
C        XREFER , YREFER , ZREFER . REFERENCE POINT FOR TRANSLATION       VCR17330
C                                                                         VCR17340
C                                                                         VCR17350
      CHARACTER*4 STOPID                                                  VCR17360
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT             VCR17370
     X ,STDEPT                                                            VCR17380
      DIMENSION D(50),DIP(50),AZIM(50)                                    VCR17390
      DIMENSION X(50),Y(50),Z(50)                                         VCR17400
      DIMENSION STOPID(50)                                                VCR17410
      DIMENSION XHOL(50),YHOL(50),ZHOL(50)                                VCR17420
      DIMENSION HOL(50,3),XP(50),YP(50),DIAM(50)                          VCR17430
      PP=3.1415926                                                        VCR17440
      REWIND 51                                                           VCR17450
      REWIND 61                                                           VCR17460
      PLELEV=PLELEV-ZREFER                                                VCR17470
      IF (PLELEV.LT.0.0) RETURN                                           VCR17480
    6 FORMAT(I1,',',I1)                                                   VCR17490
   19 FORMAT(20A4)                                                        VCR17500
   20 FORMAT(4F10.2)                                                      VCR17510
C------------------------------------------------------------------------ VCR17520
C   READ HOLE DATA                                                        VCR17530
C    99 DENOTES END OF HOLE DATA FILE                                     VCR17540
C-------------------------------------------------------------------------VCR17550
C                                                                         VCR17560
    7 READ (61.100) IHOSEQ,N1,IHOROW,IHOCOL,DIA,DIPNUM                    VCR17570
      IF (IHOSEQ.EQ.99) GO TO 8                                           VCR17580
C------------------------------------------------------------------------ VCR17590
C   INCREASE HOLE DIAMETER BY X200 FOR PLOT CLARITY                       VCR17600
```

```
C
      READ (61,101) XCOLAR,YCOLAR,ZCOLAR
C-----------------------------------------------------------------------
C        STATION INTERVAL  10 FT
C       FOLLOWING DATA REFERS TO HOLE AXIS
C-----------------------------------------------------------------------
      READ (61,101,END=8) ((X(I),Y(I),Z(I)),I=1,N1)
      CALL PTRANS(X,Y,Z,N1,XREFER,YREFER,ZREFER)
C-----------------------------------------------------------------------
C     ROTATE COORDINATES (STANGR)
C-----------------------------------------------------------------------
      DO 200 I = 1,N1
      XC =   X(I) * COS(STANGR) + Y(I) * SIN (STANGR)
      YC = - X(I) * SIN(STANGR) + Y(I) * COS (STANGR)
      X(I) = XC
      Y(I) = YC
      HOL(I,1) = X(I)
      HOL(I,2) = Y(I)
      HOL(I,3) = Z(I)
  200 CONTINUE
      DO 10 I=2,N1
      IF (HOL(I,3).LT.PLELEV) GO TO 10
      XP(1)=HOL(I,1)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,
     *1)-HOL(I,1))
      YP(1)=HOL(I,2)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,
     *2)-HOL(I,2))
      GO TO 11
   10 CONTINUE
   11 CONTINUE
      XP(2)=HOL(1,1)
      YP(2)=HOL(1,2)
      XHOL(IHOSEQ) = XP(1)
      YHOL(IHOSEQ) = YP(1)
      ZHOL(IHOSEQ) = PLELEV
      GO TO 7
    8 CONTINUE
  100 FORMAT(4I5,2F10.2)
  101 FORMAT(3F10.2)
      REWIND 51
      REWIND 61
      RETURN
      END
      SUBROUTINE PTRANS(X,Y,Z,NPNTS,XREF,YREF,ZREF)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     PURPOSE  :                                                    C
C          TO PERFORM A TRANSLATION WRT THE POINT XREF,YREF,ZREF    C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION X(50),Y(50),Z(50)
      DO 10 I=1,NPNTS
      X(I)=X(I)-XREF
      Y(I)=Y(I)-YREF
      Z(I)=Z(I)-ZREF
   10 CONTINUE
```

```
      RETURN                                                    VCR18160
      END                                                       VCR18170
```

2. LARGE DIAMETER BENCH BLAST DESIGN

```
      PROGRAM LDBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C   THIS PROGRAM DESIGNS THE BLAST FOR AN UNDERGROUND BENCH WHEN THE  C
C   POSITION OF THE HOLES AND THE CRATERING CHARACTERISTICS ARE KNOWN.C
C   THE PROGRAM GIVES AN OUTPUT INDICATING PROBLEMS DUE TO THE        C
C   DEVIATION OF THE HOLES.                                           C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       LAST EDITION :  OCTOBER 1984
C
C     VARIABLES :
C           ALENGT : LENGTH OF THE EXPLOSIVE COLUMN
C           ALOADW :
C           ALUPD(I,J)  : WEIGHT OF EXPLOSIVE IN HOLE I ,LEVEL J
C           BRTHR : KODE INDICATING IF THE HOLES ARE BREAKTHROUGH
C           BMIN  : MINIMUM BURDEN
C           BURDPF : BURDEN OF  INITIAL PATTERN
C           DCONCB : DEPTH OF CONTAINED CHARGE (CYLINDRICAL)
C           DCONCH : DEPTH OF CONTAINED CHARGE (SPHERICAL)
C           DDOUPD : CHARGE LOCATION IN THE BOREHOLE
C           DELTA  : DEPTH RATIO (SPHERICAL CHARGES)
C           DELTAB : DEPTH RATIO (CYLINDRICAL CHARGES)
C           DELTO : OPTIMUM DEPTH RATIO ( SPHERICAL )
C           DELTOB : OPTIMUM DEPTH RATIO ( CYLINDRICAL )
C           DHOLE : DIAMETER OF BOREHOLE
C           DHOLEI : INCREASED DIAMETER OF BOREHOLE (FOR GRAPHICS)
C           EFACTB : STRAIN ENERGY FACTOR FOR CYLINDRICAL CHARGES
C           EFACTR : STRAIN ENERGY FACTOR FOR SPHERICAL CHARGES
C           ELEV   : ELEVATION
C           EXPTYB : EXPLOSIVE TYPE
C           EXPTYP : EXPLOSIVE TYPE
C           IBLOCK(I,J) : IF GREATER THAN ZERO BLOCK (I,J) TAKEN
C                         OTHERWISE BLOCK IS NOT TAKEN
C           INDEXV : IF 1 VIBRATIONS EXCEED MAXIMUM
C           ISOUT  : ARRAY INDICATING SEQUENCE
C           ITIME(I,J) : ARRAY INDICATING PERIOD IN WHICH BLOCK (I,J) WAS
C                    REMOVED
C           IWK(I,J) : ARRAY FOR PLOTTING OF OUTPUT
C           MSYSUN  : 1 METRIC
C                     2 IMPERIAL
C           NDECKS  : NUMBER OF DECKS
C           NPB1    : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH
C                     RATIO CURVE (CYLINDRICAL)
C           NPB2    : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH
C                     RATIO CURVE (CYLINDRICAL)
C           NPFF    : NUMBER OF POINTS FOR THE GENERATION OF THE FREE
C                     FACE
C           NP1    : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH
C                     RATIO CURVE (SPHERICAL)
C           NP2    : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH
C                     RATIO CURVE (SPHERICAL)
C
C           NTOTAL :
```

```
C          NUMBH  :  NUMBER OF HOLES IN THE BLAST                          LLM00560
C          RHOE   : DENSITY OF THE EXPLOSIVE                               LLM00570
C          SCDEPB : SCALED DEPTH OF BURIAL FOR CYLINDRICAL CHARGES         LLM00580
C          SCDEPT : SCALED DEPTH OF BURIAL FOR SPHERICAL CHARGES           LLM00590
C          SCRAD : SCALED RADIUS FOR SPHERICAL                             LLM00600
C          SCRADB : SCALED RADIUS FOR CYLINDRICAL                          LLM00610
C          SCROP : OPTIMUM SCALED RADIUS                                   LLM00620
C          SEPDIS : SEPARATION DISTANCE BETWEEN DECKS                      LLM00630
C          TITLE : TITLE                                                   LLM00640
C          WEIGB : WEIGHT/UNIT LENGTH                                      LLM00650
C          WEIGHT : WEIGHT OF EXPLOSIVE   DECK                             LLM00660
C          WSTR  : WEIGHT STRENGTH                                         LLM00670
C          WTCHRB : WEIGHT OF CHARGE IN TESTS (CYLINDRICAL)                LLM00680
C          WTCHRG : WEIGHT OF CHARGE IN TESTS (SPHERICAL)                  LLM00690
C          WUPD(I,J)    : WEIGHT OF A DECK I, IN LEVEL J                   LLM00700
C          X(I) : X COORDINATE OF HOLE I                                   LLM00710
C          XMAX : MAX LOCAL LENGTH (USED TO DETERMINE LENGTH OF BLOCK)     LLM00720
C          XMIN : MIN LOCAL LENGTH                                         LLM00730
C          XXX(I,J) : X COORDINATE OF CHARGE I IN LEVEL J                  LLM00740
C          Y(I) : Y COORDINATE OF HOLE I                                   LLM00750
C          YMAX : MAX LOCAL WIDTH (USED TO DETERMINE LENGTH OF BLOCK)      LLM00760
C          XMIN : MIN LOCAL WIDTH                                          LLM00770
C          YYY(I,J) : Y COORDINATE OF CHARGE I IN LEVEL J                  LLM00780
C          Z(I) : Z COORDINATE OF HOLE I                                   LLM00790
C          ZZZ(I,J) : Z COORDINATE OF CHARGE I IN LEVEL J                  LLM00800
C                                                                          LLM00810
C                                                                          LLM00820
      INTEGER*4 CLTIME,TCPU                                                LLM00830
      CHARACTER*15 EXPTYP,EXPTYB                                           LLM00840
      CHARACTER*80 TITLE                                                   LLM00850
      CHARACTER*8  QDATE ,QTIM                                             LLM00860
      CHARACTER*1 KODE,IWR.BRTHR,KKODE1,KKODE1                             LLM00870
      CHARACTER*4 FT,INCH.LBS                                              LLM00880
      COMMON /RDRDT1/ DCONCH.WTCHRG.WSTRTH.SCDEPT(50),DELTA(50),           LLM00890
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)      LLM00900
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10).XPIL2(10),XPIL3(10),YPIL1(10),     LLM00910
     XYPIL2(10),YPIL3(10).ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,       LLM00920
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN.DELTAR(50),RHOE,DCONCB.SCDEPB(50),     LLM00930
     XDELTAB(50).SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN.SEPDIS           LLM00940
      COMMON/RDRDT2/ NP1.NPB1,NP2,NPB2,NUMBH.NCHK,NUMPIL,INIT(50),NCONTRLLM00950
     X .NTOTAL.MSYSUN                                                      LLM00960
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100.100)                          LLM00970
      COMMON/DXDY/ DX,DY                                                   LLM00980
      COMMON/DELT/DELTO.EFACTR;DELTOB,EFACTB                               LLM00990
      COMMON/XBLK/XB(100).YB(100)                                          LLM01000
      COMMON/ISEW/ IPER(50).KK,NR,NCH(50),NDECKS                           LLM01010
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),       LLM01020
     XSCROP.DOPTB(50)                                                      LLM01030
      COMMON/TYPE/EXPTYP.EXPTYB                                            LLM01040
      COMMON/TT/TITLE                                                      LLM01050
      COMMON/FREEF/ XFF(10),YFF(10)                                        LLM01060
      COMMON/UPDATE/XXX(50,20),YYY(50,20),ZZZ(50,20).WUPD(50,20)           LLM01070
      COMMON/ALPHA/IWK(100,100).BRTHR                                      LLM01080
      COMMON/CONTR/ XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)            LLM01090
      COMMON/REFER/ XREFER.YREFER,ZREFER.STANGR,STLENG,STWIDT.             LLM01100
```

```
      X STDEPT
       COMMON/UNALP/ LBS.INCH.FT
       DIMENSION ISOUT(50,20)
       DIMENSION TOP(50),ALUPD(50,20),DDOUPD(50,20)
C-----------------------------------------------------------------------
C  **      INITIALIZE DATA       **
C-----------------------------------------------------------------------
       NDECKS = 2
       ALENGT = 9.0
       PI = 3.14159
       DO 1901 I= 1,50
       DO 1900 J= 1,50
       IWR(I,J) = ' '
 1900  CONTINUE
 1901  CONTINUE
       DO 1902 I = 1,50
       WEIGHT(I) = 0.0
       WEIGB(I) = 0.0
       IPER(I) = 10000
 1902  CONTINUE
       DO 1904 I = 1,50
       DO 1903 J = 1,20
       ISOUT(I,J) = 0
 1903  CONTINUE
 1904  CONTINUE
       K = 1
       ICODE = 0
       ICODE1 = 0
       NCALL = 1
C-----------------------------------------------------------------------
C  **        READ DATA        **
C-----------------------------------------------------------------------
       CALL READER(NPFF)
C-----------------------------------------------------------------------
C  **  FIND LIVINGSTON'S STRAIN ENERGY FACTORS  **
C        FOR SPHERICAL AND CYLINDRICAL CHARGES
C-----------------------------------------------------------------------
       EFACTR = DCONCH / WTCHRG ** (1.0/3.0)
       EFACTB = DCONCB / WTCHRB ** (0.5)
       WRITE(6,510)
  510  FORMAT(1H1,///////////,40X,'P  R  O  G  R  A  M   L  B  B  B  './
      X                    40X,' ------------------------------',///,
      X 30X,'W R I T T E N   F O R   T H E   D E S I G N   O F   T H E',
      X //,30X,'B L A S T   I N   A N   U N D E R G R O U N D   S T O P '
      X,'E',//,30X,'U S I N G   T H E   L A R G E   D I A M E T E R   ',
      X//,30X,'B L A S T H O L E   B E N C H I N G   M E T H O D ',////)
       WRITE(6,511) TITLE
C      CALL QTIME(QDATE,QTIM,CLTIME,TCPU)
  511  FORMAT(1H1,33X,A80,//)
       WRITE(6,690) QDATE,QTIM
  690  FORMAT(//////,15X,'DATE          :  ',A8,//15X,'TIME          :  '
      X,A8//)
       WRITE(6,512) EXPTYB
  512  FORMAT(1H ,'            EXPLOSIVE USED :',A15)
       WRITE(6,513) RHOE,LBS.INCH
```

```
  513 FORMAT(1H0,'            DENSITY          :',F6.4,A4,'/CUBIC',A4)
      WRITE(6,590) DHOLE,INCH
  590 FORMAT(1H0,'            DIAMETER         :',F6.4,A4)
C-----------------------------------------------------------------------
C  **   FIND OPTIMUM DEPTH RATIO FOR  SPHERICAL CHARGES    **
C-----------------------------------------------------------------------
      DO 10 I = 1, NP1
      IF (SCDEPT(I+1).GE.SCDEPT(I)) GO TO 10
      DELTO = DELTA(I)
      IOPT = I
      GO TO 14
   10 CONTINUE
   14 CONTINUE
C-----------------------------------------------------------------------
C  **   FIND OPTIMUM DEPTH RATIO FOR CYLINDRICAL CHARGES    **
C-----------------------------------------------------------------------
      DO 2010 I = 1, NPB1
      IF (SCDEPB(I+1).GE.SCDEPB(I)) GO TO 2010
      DELTOB = DELTAB(I)
      IOPT = I
      GO TO 2014
 2010 CONTINUE
 2014 CONTINUE
C-----------------------------------------------------------------------
C  ** FIND SCALED RADIUS AT OPTIMUM DEPTH RATIO **
C-----------------------------------------------------------------------
      DO 15 I = 2, NP2
      IF((DELTAR(I).GE.DELTO).AND.(DELTAR(I-1).LE.DELTO)) GO TO 16
   15 CONTINUE
   16 SCROP = SCRAD(I-1) + (DELTO - DELTAR(I-1))/(DELTAR(I)-DELTAR(I-1))
     X *(SCRAD(I)-SCRAD(I-1))
      CALL SEPDST(IDELTA,BURDPT)
      WRITE(70,17) NUMBH,NUECKS
   17 FORMAT(1515)
      DHOLEI = DHOLE*10.0
      WRITE(70,18) DHOLEI
   18 FORMAT(F10.2)
      NTOTAL = 0
      WRITE(7,11)
   11 FORMAT('  ENTER (M) FOR MANUAL OR (A) FOR AUTOMATIC'./
     X        '              GENERATION OF THE INITIATION SEQUENCE')
      READ (7,13) KKODEI
   13 FORMAT(1A1)
 2800 CONTINUE
      NTOTAL = NTOTAL + 1
      IF (NTOTAL.EQ.1) GO TO 2845
C-----------------------------------------------------------------------
C  ** GET THE NEXT SERIES (HORIZONTAL) OF DECKS **
C-----------------------------------------------------------------------
      DO 2850 I = 1, NUMBH
      Z(I) = Z(I) - ALENGT - SEPDIS
 2850 CONTINUE
      ELEV = Z(I)
      CALL PLVIEW(ELEV,X,Y,NUMBH)
C-----------------------------------------------------------------------
```

```
C  **   GIVE INITIATION SEQUENCE   **
C-----------------------------------------------------------------------
 2845 CALL QUADM(NPFF)
      IFIRST = 0
      IF (ICODE1.EQ.1) GO TO 2899
 2855 CALL INITB(ICODE,NCALL,MAXPER,IFIRST,KKODE1)
      DO 2333 ISKA=1,NUMBH
 2333 CONTINUE
      IF (ICODE.EQ.1) GO TO 2901
 2899 IFIRST = 0
 2900 IFIRST = IFIRST + 1
      IF(ICODE1.NE.1) GO TO 2901
 4003 CALL VBRCHK( IFIRST,INDEXV,KKODE1,ICODE1 )
      IF(INDEXV.EQ.0) GO TO 2901
      DO 4001 I=1,NUMBH
      IF(IPER(I).EQ.IFIRST) IKAF=I
 4001 CONTINUE
      DO 4002 I = 1, NUMBH
      IF (IPER(I).GT.IPER(IKAF)) IPER(I) = IPER(I) + 1
 4002 CONTINUE
      IPER(IKAF) = IPER(IKAF) + 1
      GO TO 4003
 2901 DO 3000 ICR = 1,NUMBH
      IF (IPER(ICR).NE.IFIRST) GO TO 3000
      CALL CRPLT2(ICR,BURDPF,KKODE1,ICODE1)
 3000 CONTINUE
      IF (ICODE1.EQ.1) GO TO 3026
      IF (ICODE.EQ.1) GO TO 3001
      IF ( IFIRST.LT.MAXPER) GO TO 2900
      GO TO 3025
C-----------------------------------------------------------------------
C    **   INTERACTIVE PART WITH COLORGRAPHICS    **
C-----------------------------------------------------------------------
 3001 CONTINUE
      DO 4601 I=1,50
      DO 4600 J=1,50
      IF(IBLOCK(I,J).EQ.0) GO TO 4699
      IF(IWR(I,J).NE.'*') IWR(I,J)='.'
      GO TO 4600
 4699 IF(IWR(I,J).NE.'*') IWR(I,J) = ' '
 4600 CONTINUE
 4601 CONTINUE
      CALL BENGR(X,Y,XMAX,XMIN,YMAX,YMIN,NUMBH,Z(1))
      WRITE(7,4007)
 4007 FORMAT('    IS THAT SATISFACTORY  (Y/N) ?')
      READ (7,4008) KODE
 4008 FORMAT(A1)
      IF(KODE.EQ.'Y') GO TO 3009
      DO 4200 I = 1,50
      DO 4100 J = 1,50
      IF(ITIME(I,J).EQ.IFIRST) GO TO 4050
      GO TO 4100
 4050 IBLOCK(I,J) = 0
      ITIME(I,J) = 10000
 4100 CONTINUE
```

```
 4200 CONTINUE
      DO 4920 I = 1,NCD1
      IPER(IDET) = 100000
 4920 CONTINUE
      NCALL = 2
      IFIRST = IFIRST - 1
      GO TO 2855
 3009 WRITE(7,3010)
 3010 FORMAT('  WAS THIS THE LAST DELAY PERIOD ?')
      READ(7,3020) KODE
 3020 FORMAT(1A1)
      IF (KODE.EQ.'N') GO TO 2855
      ILASTP = IFIRST
 3025 WRITE(7,3024)
 3024 FORMAT('  DO YOU WANT TO USE THE SAME INITIATION SEQUENCE',/
     X        '      FOR THE SUBSEQUENT DECK LEVELS ?')
      READ(7,3023) KKODE1
 3023 FORMAT(1A1)
      IF (KKODE1.EQ.'Y') ICODE1 = 1
 3026 CONTINUE
      IF (K.EQ.1) GO TO 3028
      IF (IFIRST.LT.ILASTP) GO TO 2900
C-----------------------------------------------------------------------
C  **     UPDATE VARIABLES      **
C-----------------------------------------------------------------------
 3028 DO 3030 IU = 1,NUMBH
      WUPD(IU,K) = WEIGHT(IU)
      XXX(IU,K) = X(IU)
      YYY(IU,K) = Y(IU)
      ALUPD(IU,K) = WEIGHT(IU)/(12.0*RHOE*PI*DHOLE**2.0/4.0)
      IF(MSYSUN.EQ.1) ALUPD(IU,K) = WEIGHT(IU)/(0.1*RHOE*PI*DHOLE**2.0/
     X4.0)
      ZZZ(IU,K) = Z(IU) - ALUPD(IU,K)/2.0
 3030 CONTINUE
      CALL SEQOUT(K,ISOUT,ICODE1)
C-----------------------------------------------------------------------
C   **    WRITE PRINTOUT FOR THE LEVEL    **
C-----------------------------------------------------------------------
      DO 601 I=1,50
      DO 600 J=1,50
      IF(IBLOCK(I,J).EQ.0) GO TO 699
      IF(IWR(I,J).NE.'*') IWR(I,J)='.'
      GO TO 600
  699 IF(IWR(I,J).NE.'*') IWR(I,J) = ' '
  600 CONTINUE
  601 CONTINUE
      CALL BENGR (X,Y,XMAX,XMIN,YMAX,YMIN,NUMBH,Z(1))
      WRITE(6,496) K
  496 FORMAT('1',/////,51X,' INITIATION SEQUENCE IN LEVEL *',I2,/
     X        51X,          '-----------------------------------',//
     X 40X,'HOLE NUMBER',30X,'DELAY PERIOD',/
     X 40X,'-----------',30X,'-----------',//)
      DO 498  I=1,NUMBH
      WRITE(6,497)  I,ISOUT(I,K)
  497 FORMAT(45X,I2,40X,I2,/)
```

```
  498 CONTINUE
      WRITE(6,501)  K,LBS
  501 FORMAT(1H1,/////,51X,'WEIGHT OF EXPLOSIVE DECKS IN LEVEL #',I2,/
     X          51X,        '-----------------------------------------',//
     X 40X,'HOLE NUMBER',30X,'WEIGHT (',A4,')',/
     X 40X,'-----------',30X,'-------------',//)
       DO 503 I = 1, NUMBH
       ALOADW = WEIGHT(I)*WSTRTH/WSTR
       WRITE(6,502) I,ALOADW
  502  FORMAT(45X,I2,35X,F8.0,/)
  503  CONTINUE
      WRITE(6,7777) Z(1)
 7777 FORMAT(1H1,'                    PLAN VIEW AT DEPTH', F12.4,/
     X'                  ------------------------------------',/////)
      WRITE(6,7437) XB(1),XB(11),XB(21),XB(31),XB(41),XB(51),XB(61),
     XXB(71)
 7437 FORMAT(8X,8(F4.0,6X))
      WRITE(6,7438)
 7438 FORMAT(10X,'+---------+---------+---------+---------+---------+',
     X'---------+---------+--------')
  602 CONTINUE
      DO 7670 JP = 1,50,10
      WRITE(6,7655) YB(JP),(IWR(I,JP),I=1,50)
 7655 FORMAT(1H ,F4.0,'    + ',50A1)
      WRITE(6,7660) ((IWR(I,J),I=1,50),J=JP+1,JP+9)
 7660 FORMAT(1H ,4X  ,'    | ',50A1)
 7670 CONTINUE
      K = K + 1
      IF ( K.GT.NDECKS) GO TO 3052
      GO TO 2800
 3052 DO 7710 J = 1,NDECKS
      WRITE(70,7709) (ZZZ(I,J),I=1,NUMBH)
 7709 FORMAT(8F10.2)
 7710 CONTINUE
      DO 7720 J = 1,NDECKS
      WRITE(70,7709) (WUPD(I,J),I=1,NUMBH)
 7720 CONTINUE
      DO 7730 J = 1,NDECKS
      WRITE(70,7709) (ALUPD(I,J),I=1,NUMBH)
 7730 CONTINUE
      DO 7740 J = 1,NDECKS
      WRITE(70,17) (ISOUT(I,J),I=1,NUMBH)
 7740 CONTINUE
C-----------------------------------------------------------------------
C  **    GIVE SUMMARY OF THE REQUIRED SUPPLIES    **
C-----------------------------------------------------------------------
      TW = 0.0
      DO 9100 I = 1,NUMBH
      DO 9050 J = 1,NDECKS
      TW = TW + WUPD(I,J)
 9050 CONTINUE
 9100 CONTINUE
      WRITE(6,9101) TW,LBS
 9101 FORMAT(1H1,45X,'SUMMARY',/45X,'-------',///,25X,'TOTAL WEIGHT OF'
     X,' EXPLOSIVE  :',F10.0,A4)
```

```
      WRITE(6,9102)
 9102 FORMAT(1H0,25X,'DELAY NUMBER',40X,'QUANTITY',/
     X           ,25X,'------------',40X,'--------')
      MAXPER = 0
      DO 9105 I = 1,NUMBH
      DO 9104 J = 1,NDECKS
      IF(ISOUT(I,J).GT.500) GO TO 9104
      IF(ISOUT(I,J).GT.MAXPER) MAXPER=ISOUT(I,J)
 9104 CONTINUE
 9105 CONTINUE
      DO 9110 I = 1,MAXPER
      MM = 0
      DO 9109 J = 1,NUMBH
      DO 9199 ND = 1,NDECKS
      IF(ISOUT(J,ND).EQ.I) MM = MM + 1
 9199 CONTINUE
 9109 CONTINUE
      WRITE(6,9115) I,MM
 9115 FORMAT(1H0,30X,I2,48X,I2)
 9110 CONTINUE
      CALL CHARGL(DDOUPD)
      WRITE(6,9200)
 9200 FORMAT(1H1,45X,'INSTRUCTIONS FOR LOADING',/
     X           ,45X,'------------------------',///
     X10X,'HOLE NUMBER',10X,'CHARGE LOCATION',10X,'CHARGE WEIGHT',10X,
     X'DELAY NUMBER',/10X,'-----------',10X,'---------------',10X,
     X'-------------',10X,'------------')
      DO 9260 I = 1,NUMBH
      WRITE(6,9250) I,DDOUPD(I,1),WUPD(I,1),ISOUT(I,1)
 9250 FORMAT(1H0,15X,I2,19X,F5.0,19X,F5.0,17X,I2)
      DO 9255 J = 2,NDECKS
      WRITE(6,9252) DDOUPD(I,J),WUPD(I,J),ISOUT(I,J)
 9252 FORMAT(1H0,17X,19X,F5.0,19X,F5.0,17X,I2)
 9255 CONTINUE
 9260 CONTINUE
      STOP
      END
      SUBROUTINE CRPLT2(ICR,BURDPF,KKODEI,ICODEI)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C   PURPOSE  :                                                       C
C             TO CONSIDER THE EFFECT OF THE EXISTING CRATERS IN THE  C
C             DETONATION OF A NEW CHARGE .                           C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  :
C
C    VARIABLES :
C          BIG : A LARGE NUMBER
C          BMIN  : MINIMUM BURDEN
C          BURDPF : BURDEN OF  INITIAL PATTERN
C          DCONTB : DEPTH OF CONTAINED CHARGE
C          DELTAB : DEPTH RATIO (CYLINDRICAL CHARGES)
```

```
C          DELTBR : DEPTH RATIO                                          LLM04410
C          DEPR : DEPTH RATIO                                            LLM04420
C          DHOLE : DIAMETER OF BOREHOLE                                  LLM04430
C          DIST : DISTANCE                                               LLM04440
C          DX : X DIMENSION OF BLOCK                                     LLM04450
C          DY : Y DIMENSION OF BLOCK                                     LLM04460
C          EFACTB : STRAIN ENERGY FACTOR FOR CYLINDRICAL CHARGES         LLM04470
C          EFACTR : STRAIN ENERGY FACTOR FOR SPHERICAL CHARGES           LLM04480
C          EXPTYB : EXPLOSIVE TYPE                                       LLM04490
C          EXPTYP : EXPLOSIVE TYPE                                       LLM04500
C          IBLOCK(I,J) : IF GREATER THAN ZERO BLOCK (I,J) TAKEN          LLM04510
C                        OTHERWISE BLOCK IS NOT TAKEN                    LLM04520
C          INDEXV : IF 1 VIBRATIONS EXCEED MAXIMUM                       LLM04530
C          ISOUT  : ARRAY INDICATING SEQUENCE                            LLM04540
C          ITIME(I,J) : ARRAY INDICATING PERIOD IN WHICH BLOCK (I,J) WASLLM04550
C                  REMOVED                                               LLM04560
C          NPB1     : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH    LLM04570
C                     RATIO CURVE (CYLINDRICAL)                          LLM04580
C          NPB2     : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH   LLM04590
C                     RATIO CURVE (CYLINDRICAL)                          LLM04600
C          SCDEPB : SCALED DEPTH OF BURIAL FOR CYLINDRICAL CHARGES       LLM04610
C          SCDEPT : SCALED DEPTH OF BURIAL FOR SPHERICAL CHARGES         LLM04620
C          SCRADB : SCALED RADIUS FOR CYLINDRICAL                        LLM04630
C          SCROP : OPTIMUM SCALED RADIUS                                 LLM04640
C          WEIGB : WEIGHT/UNIT LENGTH                                    LLM04650
C          WEIGHT : WEIGHT OF EXPLOSIVE   DECK                           LLM04660
C          WSTR  : WEIGHT STRENGTH                                       LLM04670
C          X(I) : X COORDINATE OF HOLE I                                 LLM04680
C          XMAX : MAX LOCAL LENGTH (USED TO DETERMINE LENGTH OF BLOCK)   LLM04690
C          XMIN : MIN LOCAL LENGTH                                       LLM04700
C          Y(I) : Y COORDINATE OF HOLE I                                 LLM04710
C          YMAX : MAX LOCAL WIDTH (USED TO DETERMINE LENGTH OF BLOCK)    LLM04720
C          XMIN : MIN LOCAL WIDTH                                        LLM04730
C          Z(I) : Z COORDINATE OF HOLE I                                 LLM04740
C                                                                        LLM04750
C     XS,YS,ZS : ARRAYS DEFINING A CLOSED POLYGON                        LLM04760
C-----------------------------------------------------------------------LLM04770
      CHARACTER*1 KKODEI                                                 LLM04780
      COMMON/IBLK/ IBLOCK(100,100), ITIME(100,100)                       LLM04790
      COMMON/XBLK/ XB(100),YB(100)                                       LLM04800
      COMMON/DXDY/ DX,DY                                                 LLM04810
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),         LLM04820
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)    LLM04830
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),   LLM04840
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,     LLM04850
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),   LLM04860
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS         LLM04870
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTRLLM04880
     X , NTOTAL                                                          LLM04890
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB                             LLM04900
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS                         LLM04910
      COMMON/CHARGE/ WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),    LLM04920
     XSCROP,DOPTB(50)                                                    LLM04930
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,           LLM04940
     X STDEPT                                                            LLM04950
```

```
      DIMENSION BURMIN(50),XS(10),YS(10)
C-----------------------------------------------------------------------
C  **  INITIALIZE VARIABLES  **
C-----------------------------------------------------------------------
      BIG = 10000.0
      DO 10 INV = 1,10
      XS(INV) = 0.0
      YS(INV) = 0.0
   10 CONTINUE
      L = IPER(ICR) - 1
      L1 = IPER(ICR)
      CALL BRDCHK(L,ICR,BURDPF,KKODE1,ICODE1)
      IF (WEIGHT(ICR).LT.0.2)  RETURN
C-----------------------------------------------------------------------
C  ** IF BMIN GREATER THAN DEPTH OF CONTAINED CHARGE, POINT DOES NOT
C          AFFECT DETONATION
C-----------------------------------------------------------------------
      DCONTB = EFACTB * WEIGB(ICR)**0.5
      IF (BMIN.GT.DCONTB) GO TO 2222
C-----------------------------------------------------------------------
C  **   FIND CRATER RADIUS FOR MINIMUM BURDEN   **
C-----------------------------------------------------------------------
      DEPR = BMIN / DCONTB
C-----------------------------------------------------------------------
C  **  FIND SCALED RADIUS AT THIS DEPTH RATIO  **
C-----------------------------------------------------------------------
      DO 40  MI = 2, NPB2
      IF ((DELTBR(MI).GE.DEPR).AND.(DELTBR(MI-1).LE.DEPR)) GO TO 35
   40 CONTINUE
   35 SCR = SCRADB(MI-1) + (DEPR - DELTBR(MI-1))/(DELTBR(MI) -
     XDELTBR(MI-1))*(SCRADB(MI)-SCRADB(MI-1))
      RADIUS = SCR * WEIGB(ICR)**0.5
C-----------------------------------------------------------------------
C  **   CALCULATE SLANT DISTANCE   **
C-----------------------------------------------------------------------
      SLANTD = (RADIUS**2.0 + BMIN**2.0)**0.5
      DO 2020 II = 2, 49
      DO 2010 JJ = 2, 49
C-----------------------------------------------------------------------
C  **  CHECK IF POINT (II,JJ,KKK) IN PERIOD L IS INSIDE THE MASS
C       AND THEREFORE IT DOES NOT  AFFECT THE DETONATION    **
C-----------------------------------------------------------------------
      IF(ITIME(II,JJ).GT.L)GO TO 2000
      III=II+1
      JJ1=JJ+1
      IIM=II-1
      JJM=JJ-1
C-----------------------------------------------------------------------
C  ** POINT MAY AFFECT DETONATION. CALCULATE DISTANCE BETWEEN POINT
C     AND THE CENTRE OF THE CHARGE.
C-----------------------------------------------------------------------
      DIST = (ABS(Y(ICR)-YB(JJ))**2.0+ABS(X(ICR)-XB(II))**2.0)**0.5
      IF(DIST.GT.SLANTD) GO TO 2010
C-----------------------------------------------------------------------
C ** BLOCKS FURTHER THAN THE SLANT DISTANCE DO NOT AFFECT DETONATION **
```

```
C-----------------------------------------------------------------------
      XS(1)=XB(II)
      YS(1)=YB(JJ)
      DO 1025 I1=2,49
      DO 1020 J1=2,49
      IF((I1.EQ.I1).AND.(J1.EQ.JJ)) GO TO 1020
      IF(ITIME(I1,J1).GT.L) GO TO 1020
      II1=I1+1
      JJ1=J1+1
      IIM=I1-1
      JJM=J1-1
      IF((IBLOCK(II1,J1).NE.0).AND.(IBLOCK(I1,JJ1).NE.0).AND.
     X(IBLOCK(IIM,J1).NE.0).AND.(IBLOCK(I1,JJM).NE.0).AND.
     X(ITIME(II1,J1).LE.L).AND.(ITIME(I1,JJ1).LE.L).AND.
     X(ITIME(IIM,J1).LE.L).AND.(ITIME(I1,JJM).LE.L))
     X GO TO 1020
      DIST1=(ABS(XB(I1)-X(ICR))**2.0+ABS(YB(J1)-Y(ICR))**2.0)**0.5
      IF(DIST1.LE.(DIST-DX/2.0).OR.(DIST1.GE.(DIST+DX/2.0))) GO TO 1020
      XS(2) = XB(I1)
      YS(2) = YB(J1)
      GO TO 1030
 1020 CONTINUE
 1025 CONTINUE
      GO TO 2010
 1030 XS(3)=X(ICR)
      DO 3047  MI = 2, NPB1
      IF ((DELTAB(NPB1).LT.DEPR)) GO TO 3032
      IF ((DELTAB(MI).GE.DEPR).AND.(DELTAB(MI-1).LE.DEPR)) GO TO 3035
      IF (DELTAB(1).GE.DEPR) GO TO 3034
 3047 CONTINUE
      STOP
 3032 SCD = SCDEPB(NPB1)-(DEPR-DELTAB(NPB1))/(1-DELTAB(NPB1))*
     XSCDEPB(NPB1)
      GO TO 3036
 3034 SCD = (DEPR /DELTAB(1))*SCDEPT(1)
      GO TO 3036
 3035 SCD = SCDEPB(MI-1) + (DEPR - DELTAB(MI-1))/(DELTAB(MI) -
     XDELTAB(MI-1))*(SCDEPB(MI)-SCDEPB(MI-1))
 3036 DEPTHB = SCD * WEIGB(ICR)**0.5
      YS(3)=Y(ICR)
      DO 1045 II1=1,50
      DO 1040 JJ1=1,50
      IF(IBLOCK(II1,JJ1).NE.0) GO TO 1040
      XSO=XB(II1)
      YSO=YB(JJ1)
      CALL INSIDE(3,XS,YS,XSO,YSO,KKODE)
      IF(KKODE.EQ.0) GO TO 1040
      IBLOCK(II1,JJ1) = 1
      ITIME(II1,JJ1) = IPER(ICR)
 1040 CONTINUE
 1045 CONTINUE
 2000 CONTINUE
 2010 CONTINUE
 2020 CONTINUE
 2222 RETURN
```

```
      END                                                               LLM06060
      SUBROUTINE VBRCHK (IFIRST,INDEXV,KKODEI,ICODE1 )                  LLM06070
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLLM06080
C                                                                      LLM06090
C  PURPOSE :                                                           LLM06100
C            TO EXAMINE THE VIBRATIONS RESULTING FROM THE DETONATIONS  LLM06110
C            OF SINGLE AND MULTIPLE CHARGES.                           LLM06120
C                                                                      LLM06130
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLLM06140
C                                                                      LLM06150
C    VARIABLES :                                                       LLM06160
C          AKFAC : K FACTOR FOR VIBRATIONS                             LLM06170
C          ALENGT : LENGTH OF THE EXPLOSIVE COLUMN                     LLM06180
C          ALOAD : WEIGHT OF EXPLOSIVE IN CHARGE                       LLM06190
C          AMFAC : M FACTOR FOR VIBRATIONS                             LLM06200
C          DHOLE : DIAMETER OF BOREHOLE                                LLM06210
C          DIST,DISTM,DISTP : DISTANCE                                 LLM06220
C          INDEXV : IF 1 VIBRATIONS EXCEED MAXIMUM                     LLM06230
C          IPER(I) : DELAY PERIOD (LOCAL)                              LLM06240
C          ITIME(I,J) : ARRAY INDICATING PERIOD IN WHICH BLOCK (I,J) WASLLM06250
C                   REMOVED                                            LLM06260
C          MSYSUN  : 1 METRIC                                          LLM06270
C                    2 IMPERIAL                                        LLM06280
C          NCHK : NUMBER OF POINTS FOR VIBRATION CHECK                 LLM06290
C          NDECKS  : NUMBER OF DECKS                                   LLM06300
C                                                                      LLM06310
C          NUMBH  :  NUMBER OF HOLES IN THE BLAST                      LLM06320
C          PPVALL : MAX PPV FOR POINTS                                 LLM06330
C          PPVPAL : MAX PPV FOR PILLARS                                LLM06340
C          RCR : CRATER RADIUS                                         LLM06350
C          SEPDIS : SEPARATION DISTANCE BETWEEN DECKS                  LLM06360
C          WEIGB : WEIGHT/UNIT LENGTH                                  LLM06370
C          WEIGHT : WEIGHT OF EXPLOSIVE   DECK                         LLM06380
C          WSTR  : WEIGHT STRENGTH                                     LLM06390
C          WTCHRB : WEIGHT OF CHARGE IN TESTS (CYLINDRICAL)            LLM06400
C          WTCHRG : WEIGHT OF CHARGE IN TESTS (SPHERICAL)              LLM06410
C          X(I) : X COORDINATE OF HOLE I                               LLM06420
C          Y(I) : Y COORDINATE OF HOLE I                               LLM06430
C                                                                      LLM06440
      CHARACTER*1 KKODE1,SEQ                                           LLM06450
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),       LLM06460
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)  LLM06470
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10), LLM06480
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,   LLM06490
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50), LLM06500
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS       LLM06510
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTRLLM06520
     X ,NTOTAL,MSYSUN                                                  LLM06530
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),   LLM06540
     XSCROP,DOPTB(50)                                                  LLM06550
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB                           LLM06560
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS                       LLM06570
      COMMON/CONTR/XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)         LLM06580
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,         LLM06590
     X STDEPT                                                          LLM06600
```

```
      DIMENSION ALOAD(50),PPV(10),PPVPIL(10),WEIGMX(50),ID(50),TOP(50)  LLM06610
C-----------------------------------------------------------------------  LLM06620
C  **  INITIALIZE VARIABLES  **                                          LLM06630
C-----------------------------------------------------------------------  LLM06640
      DO 2 I = 1,50                                                      LLM06650
      ID(I) = 0                                                          LLM06660
    2 CONTINUE                                                           LLM06670
      WEIGMX(1) = 100000.0                                               LLM06680
      INDEXV = 0                                                         LLM06690
C-----------------------------------------------------------------------  LLM06700
C  **  FIND CHARGES DETONATING IN THE SAME PERIOD  **                    LLM06710
C-----------------------------------------------------------------------  LLM06720
      KK = 0                                                             LLM06730
      DO 100  IA = 1, NUMBH                                              LLM06740
      IF (IPER(IA).NE.IFIRST) GO TO 100                                  LLM06750
C-----------------------------------------------------------------------  LLM06760
C  **          LOAD HOLE           **                                   LLM06770
C-----------------------------------------------------------------------  LLM06780
      PI = 3.14159                                                       LLM06790
      ALOAD(IA) = (12.0*ALENGT * PI * DHOLE ** 2.0 / 4.0) * RHOE         LLM06800
      IF(MSYSUN.EQ.1) ALOAD(IA) = (100.0*ALENGT*PI*DHOLE**2.0/4.0)*RHOE  LLM06810
     X/1000.0                                                            LLM06820
C-----------------------------------------------------------------------  LLM06830
C  **  CALCULATE EQUIVALENT WEIGHT OF EXPLOSIVE  **                      LLM06840
C-----------------------------------------------------------------------  LLM06850
      WEIGHT(IA) = ALOAD(IA) * WSTR / WSTRTH                             LLM06860
      WEIGB(IA) = WEIGHT(IA)/ALENGT                                      LLM06870
      IF (NTOTAL.NE.NDECKS) GO TO 20                                     LLM06880
      CALL TOPH(IA,TOP(IA))                                              LLM06890
      ALOAD(IA) = (12.0*(SEPDIS+ALENGT-TOP(IA))*PI*DHOLE**2.0/4.0)*RHOE  LLM06900
      IF(MSYSUN.EQ.1) ALOAD(IA) = ALOAD(IA)/12.0 * 0.1                   LLM06910
      IF (ALOAD(IA).GT.0.0) GO TO 11                                     LLM06920
      ALOAD(IA) = 0.0                                                    LLM06930
      WEIGHT(IA) = 0.0                                                   LLM06940
      GO TO 200                                                          LLM06950
   11 WEIGHT(IA) = ALOAD(IA) * WSTR/WSTRTH                               LLM06960
C-----------------------------------------------------------------------  LLM06970
C  **   CHECK VIBRATION LEVELS   **                                      LLM06980
C-----------------------------------------------------------------------  LLM06990
   20 IF(NCHK.EQ.0) GO TO 32                                             LLM07000
C-----------------------------------------------------------------------  LLM07010
C  **  CALCULATE DISTANCES FROM POINTS                                   LLM07020
C-----------------------------------------------------------------------  LLM07030
      DO 30 I = 1, NCHK                                                  LLM07040
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),X(IA),Y(IA),Z(IA),DIST)     LLM07050
C-----------------------------------------------------------------------  LLM07060
C  **  CALCULATE PPV AT THIS POINT  **                                   LLM07070
C-----------------------------------------------------------------------  LLM07080
      PPV(I) = AKFAC*(DIST/WEIGHT(IA)**(1.0/2.0))**AMFAC                 LLM07090
C-----------------------------------------------------------------------  LLM07100
C  ** FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT **                   LLM07110
C-----------------------------------------------------------------------  LLM07120
      WEIGMX(I) = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DIST**2.0               LLM07130
   30 CONTINUE                                                           LLM07140
C-----------------------------------------------------------------------  LLM07150
```

```
C  ** CALCULATE DISTANCE FROM PILLARS **                                LLM07160
C-----------------------------------------------------------------------LLM07170
   32 IF (NUMPIL.EQ.0) GO TO 52                                          LLM07180
      DO 50 I = 1. NUMPIL                                                LLM07190
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL2(I),YPIL2(I),ZPIL2(I),LLM07200
     XXPIL3(I),YPIL3(I),ZPIL3(I),X(IA),Y(IA),Z(IA),DISTP)                LLM07210
C-----------------------------------------------------------------------LLM07220
C  ** CALCULATE PPV AT THIS POINT **                                    LLM07230
C-----------------------------------------------------------------------LLM07240
      PPVPIL(I) = AKFAC * (DISTP/WEIGHT(IA)**(1.0/2.0))**                LLM07250
     X AMFAC                                                             LLM07260
C-----------------------------------------------------------------------LLM07270
C  ** FIND MAXIMUM WEIGHT BECAUSE OF THIS PARTICULAR PILLAR **          LLM07280
C-----------------------------------------------------------------------LLM07290
      WEIGMX(NCHK+1) = (AKFAC/PPVPAL)**(2.0/AMFAC)*DISTP**2.0            LLM07300
   50 CONTINUE                                                           LLM07310
C-----------------------------------------------------------------------LLM07320
C  ** FIND MINIMUM ACCEPTABLE WEIGHTS **                                LLM07330
C-----------------------------------------------------------------------LLM07340
   52 WMIN = WEIGMX(1)                                                   LLM07350
      NN = NCHK + NUMPIL                                                 LLM07360
      IF (NN.EQ.0) GO TO  99                                             LLM07370
      DO 80 I = 1, NN                                                    LLM07380
      IF (WEIGMX(I).LT.WMIN) WMIN=WEIGMX(I)                              LLM07390
   80 CONTINUE                                                           LLM07400
      IF (WEIGHT(IA).LE.WMIN) GO TO 99                                   LLM07410
      WEIGHT(IA) = WMIN                                                  LLM07420
   99 KK = KK + 1                                                        LLM07430
      INIT(KK) = IA                                                      LLM07440
  100 CONTINUE                                                           LLM07450
      IF(NN.EQ.0) GO TO 200                                              LLM07460
C-----------------------------------------------------------------------LLM07470
C      CHECK VIBRATIONS DUE TO THE SIMULTANEOUS  DETONATION OF          LLM07480
C                 MANY CHARGES                                          LLM07490
C-----------------------------------------------------------------------LLM07500
C                                                                       LLM07510
C-----------------------------------------------------------------------LLM07520
C  ** FIND AN AVERAGE POSITION OF THE CONCENTRATED CHARGE **            LLM07530
C-----------------------------------------------------------------------LLM07540
      XM = 0.0                                                           LLM07550
      IF(KK.EQ.1) GO TO 200                                              LLM07560
      YM = 0.0                                                           LLM07570
      ZM = 0.0                                                           LLM07580
      WM = 0.0                                                           LLM07590
      DO 110  I = 1,KK                                                   LLM07600
      IM = INIT(I)                                                       LLM07610
      XM = XM + X(I)*WEIGHT(IM)                                          LLM07620
      YM = YM + Y(I)*WEIGHT(IM)                                          LLM07630
      ZM = ZM + Z(I)*WEIGHT(IM)                                          LLM07640
      WM = WM + WEIGHT(IM)                                               LLM07650
  110 CONTINUE                                                           LLM07660
      XM = XM / WM                                                       LLM07670
      YM = YM / WM                                                       LLM07680
      ZM = ZM / WM                                                       LLM07690
C-----------------------------------------------------------------------LLM07700
```

```
C  **  CALCULATE DISTANCE FROM POINTS  **                                LLM07710
C-----------------------------------------------------------------------LLM07720
      IF (NCHK.EQ.0) GO TO 131                                           LLM07730
      DO 130 I = 1, NCHK                                                 LLM07740
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),XM,YM,ZM,DISTM)             LLM07750
C-----------------------------------------------------------------------LLM07760
C  **   CALCULATE PPV AT THIS POINT   **                                LLM07770
C-----------------------------------------------------------------------LLM07780
      PPV(I) = AKFAC*(DISTM/WM*(1.0/2.0)) ** AMFAC                       LLM07790
C-----------------------------------------------------------------------LLM07800
C  **  FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT  **                LLM07810
C-----------------------------------------------------------------------LLM07820
      WEIGMX(I) = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DISTM**2.0              LLM07830
  130 CONTINUE                                                           LLM07840
C-----------------------------------------------------------------------LLM07850
C  **       SAME FOR PILLARS                                             LLM07860
C-----------------------------------------------------------------------LLM07870
  131 IF (NUMPIL.EQ.0) GO TO 151                                         LLM07880
      DO 150 I = 1, NUMPIL                                               LLM07890
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL2(I),YPIL2(I),ZPIL2(I).LLM07900
     X XPIL3(I),YPIL3(I),ZPIL3(I),XM,YM,ZM,DISTM)                        LLM07910
      PPVPIL(I) = AKFAC*((DISTM)/WM**(1.0/2.0))**AMFAC                   LLM07920
      WEIGMX(NCHK+I) = (AKFAC/PPVPAL)**(2.0/AMFAC)*DISTM**2.0            LLM07930
  150 CONTINUE                                                           LLM07940
C-----------------------------------------------------------------------LLM07950
C  **    FIND MINIMUM OF ACCEPTABLE WEIGHTS    **                        LLM07960
C-----------------------------------------------------------------------LLM07970
  151 WMIN = 2000000.0                                                   LLM07980
      NN = NCHK + NUMPIL                                                 LLM07990
      IF (NN.EQ.0) GO TO 200                                             LLM08000
      DO 180 I = 1,NN                                                    LLM08010
      IF(WEIGMX(I).LT.WMIN) WMIN = WEIGMX(I)                             LLM08020
  180 CONTINUE                                                           LLM08030
      IF (WM.LE.WMIN) GO TO 200                                          LLM08040
      IF (KKODEI.NE.'N') GO TO 192                                       LLM08050
      IF ( ICODE1.EQ.1 )GO TO 199                                        LLM08060
      WRITE(7,187)                                                       LLM08070
  187 FORMAT('  THE NUMBER OF CHARGES DETONATING IN THIS PERIOD IS',/    LLM08080
     X'  TOO LARGE RESULTING IN UNACCEPTABLE VIBRATIONS.',/              LLM08090
     X'  DO YOU WANT TO CHANGE THE SEQUENCING OF THE CHARGES ? (Y/N)')   LLM08100
      READ (7,188) SEQ                                                   LLM08110
  188 FORMAT(1A1)                                                        LLM08120
      INDEXV = 1                                                         LLM08130
      IF(SEQ.EQ.'Y') GO TO 191                                           LLM08140
      INDEXV = 0                                                         LLM08150
      GO TO 200                                                          LLM08160
  192 INDEXV = 1                                                         LLM08170
  191 DO 190 I = 1,NUMBH                                                 LLM08180
      IF(IPER(I).NE.IFIRST) GO TO 190                                    LLM08190
      WEIGHT(I) = 0.0                                                    LLM08200
      WEIGB(I) = 0.0                                                     LLM08210
      IPER(I) = 100000                                                   LLM08220
  190 CONTINUE                                                           LLM08230
      GO TO 200                                                          LLM08240
  199 INDEXV = 1                                                         LLM08250
```

```
      DO 2190 I = 1,NUMBH                                               LLM08260
      IF(IPER(I).NE.IFIRST) GO TO 2190                                  LLM08270
      WEIGHT(I) = 0.0                                                   LLM08280
      WEIGB(I) = 0.0                                                    LLM08290
 2190 CONTINUE                                                          LLM08300
  200 RETURN                                                            LLM08310
      END                                                               LLM08320
      SUBROUTINE QUADM(NPEF)                                            LLM08330
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLLM08340
C                                                                      CLLM08350
C   PURPOSE :                                                          CLLM08360
C             TO DESIGN THE INITIAL BLOCK MODEL                        CLLM08370
C                                                                      CLLM08380
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLLM08390
C                                                                      LLM08400
C   VARIABLES :                                                        LLM08410
C    ***   INPUT  ***                                                  LLM08420
C   XMAX : MAX DIMENSION OF THE BLAST IN THE X DIRECTION               LLM08430
C   XMIN : MIN     "        "  "     "    "  "  "    "                 LLM08440
C   YMAX : MAX     "        "  "     "    "  "  Y    "                 LLM08450
C   YMIN : MIN     "        "  "     "    "  "  "    "                 LLM08460
C   ZMAX : MAX     "        "  "     "    "  "  Z    "                 LLM08470
C   ZMIN : MIN     "        "  "     "    "  "  "    "                 LLM08480
C   MAXP : MAXIMUM PERIOD (IPER(NUMBH))                                LLM08490
C                                                                      LLM08500
C    ***  OUTPUT  ***                                                  LLM08510
C   DX : DISTANCE BETWEEN TWO CONSEQUTIVE POINTS IN THE X DIRECTION    LLM08520
C   DY :    "          "     "        "          "    "  "  Y     "   LLM08530
      CHARACTER*1 IWR,BRTHR                                             LLM08540
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),        LLM08550
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)   LLM08560
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),  LLM08570
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,    LLM08580
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),  LLM08590
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS        LLM08600
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTRLLM08610
     X ,NTOTAL,MSYSUN                                                   LLM08620
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)                       LLM08630
      COMMON/XBLK/ XB(100),YB(100)                                      LLM08640
      COMMON/DXDY/ DX,DY                                                LLM08650
      COMMON/FREEF/ XFF(10),YFF(10)                                     LLM08660
      COMMON/ALPHA/IWR(100,100),BRTHR                                   LLM08670
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,          LLM08680
     X STDEPT                                                           LLM08690
      IBIG = 3000                                                       LLM08700
      DX = (XMAX - XMIN)/50.0                                           LLM08710
      DY = (YMAX - YMIN)/50.0                                           LLM08720
      DO 10 I=1,50                                                      LLM08730
      DO  9 J=1,50                                                      LLM08740
      IBLOCK(I,J) = 0                                                   LLM08750
      ITIME(I,J) = IBIG                                                 LLM08760
      YB(J) = YMIN + DY*FLOAT(J-1) + DY/2.0                             LLM08770
    9 CONTINUE                                                          LLM08780
      XB(I) = XMIN + DX*FLOAT(I-1) + DX/2.0                             LLM08790
   10 CONTINUE                                                          LLM08800
```

```
      DO 1010 I=1,50                                                    LLM08810
      DO 1000 J=1,50                                                    LLM08820
      DO 990  IA = 1,NUMBR                                              LLM08830
      IF((X(IA).GE.(XB(I)-DX/2.0)).AND.(X(IA).LT.(XB(I)+DX/2.0)).AND.   LLM08840
     X(Y(IA).GE.(YB(J)-DY/2.0)).AND.(Y(IA).LT.(YB(J)+DY/2.0)))          LLM08850
     X IWR(I,J) = '*'                                                   LLM08860
  990 CONTINUE                                                          LLM08870
 1000 CONTINUE                                                          LLM08880
 1010 CONTINUE                                                          LLM08890
      DO 2020 I=1,50                                                    LLM08900
      DO 2010 J=1,50                                                    LLM08910
      CALL INSIDE(NPEF,XEF,YEF,XB(I),YB(J),KKODE)                       LLM08920
      IF(KKODE.EQ.0) GO TO 2010                                         LLM08930
      IBLOCK(I,J) = 1                                                   LLM08940
      ITIME(I,J) = 0                                                    LLM08950
 2010 CONTINUE                                                          LLM08960
 2020 CONTINUE                                                          LLM08970
      RETURN                                                            LLM08980
      END                                                               LLM08990
      SUBROUTINE INSIDE (NPN,X,Y,X0,Y0,KODE)                            LLM09000
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC LLM09010
C                                                                     C LLM09020
C   PURPOSE  :                                                        C LLM09030
C       TO DETERMINE IF A POINT LIES WITHIN A CLOSED                  C LLM09040
C    BOUNDARY                                                         C LLM09050
C                                                                     C LLM09060
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC LLM09070
C                                                                     LLM09080
C  VARIABLES :                                                        LLM09090
C    NPN : THE NUMBER OF POINTS DEFINING THE CLOSED POLYGON           LLM09100
C    X,Y : THE ARRAYS WITHE THE ABOVE POINT COORDINATES               LLM09110
C    KODE: =-1 THE POINT IS INSIDE BOUNDARY                           LLM09120
C    KODE: =0  THE POINT IS OUTSIDE BOUNDARY                          LLM09130
C    KODE: =1  THE POINT IS ON THE INSIDE BOUNDARY  IN VECTOR (1-1,1) LLM09140
C                                                                     LLM09150
      DIMENSION X(10),Y(10)                                             LLM09160
C-----------------------------------------------------------------------LLM09170
C  MINIMUM POINTS 3  MAXIMUM POINTS 999                                LLM09180
C-----------------------------------------------------------------------LLM09190
      IF (NPN.LE.2.OR.NPN.GT.1999)  RETURN                              LLM09200
      PP=3.141592                                                       LLM09210
      KODE=-2                                                           LLM09220
C                                                                       LLM09230
       X1=X(NPN)-X0                                                     LLM09240
       Y1=Y(NPN)-Y0                                                     LLM09250
       SUM=0.0                                                          LLM09260
       NTELOS=NPN                                                       LLM09270
       DO 401 KPN=1,NTELOS                                              LLM09280
       ANGL=0.0                                                         LLM09290
       X2=X(KPN)-X0                                                     LLM09300
       Y2=Y(KPN)-Y0                                                     LLM09310
       IF (X2.EQ.X1.AND.Y2.EQ.Y1) GO TO 401                             LLM09320
       ANOM=X1*Y2-X2*Y1                                                 LLM09330
       DENOM=X1*X2+Y1*Y2                                                LLM09340
       IF (DENOM.NE.0.0) ANGL=ATAN(ANOM/DENOM)                          LLM09350
```

```
      IF (DENOM.EQ.0.0) ANGL=PP/2.
      IF (DENOM.EQ.0.0.AND.ANOM.LT.0.0) ANGL=-PP/2.
      X1=X2
      Y1=Y2
      IF (DENOM.LT.0.0.AND.ANGL.EQ.0.0) KODE=KPN
      IF (KODE.EQ.KPN) RETURN
C-----------------------------------------------------------------------
C   IF ANGL=180. THE POINT FALLS ON THE AREA BOUNDARY
C-----------------------------------------------------------------------
      IF (ANGL.LT.0.0.AND.ANOM.GT.0.0) ANGL=ANGL+PP
      IF (ANGL.GT.0.0.AND.ANOM.LT.0.0) ANGL=ANGL-PP
      SUM=SUM+ANGL
  401 CONTINUE
      P22 = 2.0* PP
      P44 = 4.0* PP
      P66 = 6.0* PP
C-----------------------------------------------------------------------
C   IF SUM =0. THE POINT FALLS OUTSIDE THE AREA BOUNDARY
C-----------------------------------------------------------------------
      IF (ABS(SUM).LT.0.3) KODE=0
      IF (ABS(SUM).LT.0.3) RETURN
C-----------------------------------------------------------------------
C   IF SUM =2*PP THE POINT FALLS INSIDE THE AREA BOUNDARY
C-----------------------------------------------------------------------
      IF (ABS(SUM).GT.(2.0*PP-0.3)) KODE=-1
      IF (KODE.EQ.(-1)) RETURN
C-----------------------------------------------------------------------
C   SHOULD NOT GET HERE. ENTER AN ERROR CODE
C-----------------------------------------------------------------------
      KODE = -2
      RETURN
      END
      SUBROUTINE BRDCHK(L,ICR,BURDPF,KKODEI,ICODE1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C   PURPOSE  :                                                          C
C              TO CHECK BURDENS OF NEW CHARGE TO THE PREVIOUS           C
C               CRATERS AND DETERMINE IF THE CHARGE IS IN               C
C               THE FLYROCK REGION.                                     C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES :
C          ALENGT : LENGTH OF THE EXPLOSIVE COLUMN
C          BCHK1 : BURDEN
C          BCHECK : BURDEN
C          BRTHR : KODE INDICATING IF THE HOLES ARE BREAKTHROUGH
C          BMIN  : MINIMUM BURDEN
C          BURDPF : BURDEN OF  INITIAL PATTERN
C          DELTO : OPTIMUM DEPTH RATIO ( SPHERICAL )
C          DELTOB : OPTIMUM DEPTH RATIO ( CYLINDRICAL )
C          DHOLE : DIAMETER OF BOREHOLE
C          EFACTB : STRAIN ENERGY FACTOR FOR CYLINDRICAL CHARGES
C          EFACTR : STRAIN ENERGY FACTOR FOR SPHERICAL CHARGES
C          ELEV   : ELEVATION
```

```
C          EXPTYB : EXPLOSIVE TYPE                                      LLM09910
C          EXPTYP : EXPLOSIVE TYPE                                      LLM09920
C          INDEXV : IF 1 VIBRATIONS EXCEED MAXIMUM                      LLM09930
C          ITIME(I,J) : ARRAY INDICATING PERIOD IN WHICH BLOCK (I,J) WASLLM09940
C                   REMOVED                                             LLM09950
C          MSYSUN  : 1 METRIC                                           LLM09960
C                    2 IMPERIAL                                         LLM09970
C          RATIO :  RATIO BURDEN TO ADJACENT FREE FACE/OPTIMUM BURDEN   LLM09980
C          SEPDIS : SEPARATION DISTANCE BETWEEN DECKS                   LLM09990
C          WEIGB : WEIGHT/UNIT LENGTH                                   LLM10000
C          WEIGHT : WEIGHT OF EXPLOSIVE   DECK                          LLM10010
C          X(I) : X COORDINATE OF HOLE I                                LLM10020
C          XB(I) : X COORDINATE OF POINT I (BLOCK I,J)                  LLM10030
C          Y(I) : Y COORDINATE OF HOLE I                                LLM10040
C          YB(J): Y COORDINATE OF POINT J (BLOCK I,J)                   LLM10050
C-----------------------------------------------------------------------LLM10060
      CHARACTER*1 KKODE1                                                LLM10070
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),        LLM10080
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)   LLM10090
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),  LLM10100
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,    LLM10110
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),  LLM10120
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS        LLM10130
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)                       LLM10140
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB                            LLM10150
      COMMON/XBLK/XB(100),YB(100)                                       LLM10160
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS                        LLM10170
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),    LLM10180
     XSCROP,DOPTB(50)                                                   LLM10190
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCUNTRLLM10200
     X ,NTOTAL,MSYSUN                                                   LLM10210
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,          LLM10220
     X STDEPT                                                           LLM10230
      DIMENSION  IPREV(50) , DIST(50,50)                                LLM10240
C-----------------------------------------------------------------------LLM10250
C        ***    INITIALIZE   ***                                        LLM10260
C-----------------------------------------------------------------------LLM10270
      BMIN = 100000.0                                                   LLM10280
      DO 100 I= 1,50                                                    LLM10290
      DO 90 J=1,50                                                      LLM10300
      IF(ITIME(I,J).GT.IPER(ICR)) GO TO 90                              LLM10310
      DISTAN=(ABS(X(ICR)-XB(I))**2.0+ABS(Y(ICR)-YB(J))**2.0)**0.5       LLM10320
      IF (DISTAN.GE.BMIN) GO TO 90                                      LLM10330
      BMIN = DISTAN                                                     LLM10340
   90 CONTINUE                                                          LLM10350
  100 CONTINUE                                                          LLM10360
      BCHECK = 1.1*BURDPF                                               LLM10370
      IF (BMIN.LT.BCHECK) GO TO 220                                     LLM10380
C-----------------------------------------------------------------------LLM10390
C  **     CHECK SEPARATION DISTANCE       **                            LLM10400
C-----------------------------------------------------------------------LLM10410
      SEPMIN = DHOLE                                                    LLM10420
      SEPDIF = SEPDIS - SEPMIN                                          LLM10430
      IF (SEPDIF.LE.0.05) GO TO 150                                     LLM10440
      RATIO = SEPDIF/SEPDIS                                             LLM10450
```

```
      BCHK1 = 1.3*BURDPF
      IF (BMIN.GT.BCHK1) GO TO 250
      IF(RATIO.GE.0.2) GO TO 230
      WEIGHT(ICR) = RATIO*WEIGHT(ICR)
      GO TO 300
  230 WEIGHT(ICR) =1.2*WEIGHT(ICR)
      GO TO 300
  250 BCHK1 = 1.5*BURDPF
      IF(BMIN.GT.BCHCK1) GO TO 350
      IF(RATIO.GE.0.4) GO TO 260
      WEIGHT(ICR) = RATIO*WEIGHT(ICR)
      GO TO 300
  260 WEIGHT(ICR) = 1.4*WEIGHT(ICR)
      GO TO 300
  350 IF(RATIO.GE.0.6) GO TO 360
      WEIGHT(ICR) = RATIO*WEIGHT(ICR)
      GO TO 300
  360 WEIGHT(ICR) = 1.6*WEIGHT(ICR)
C-----------------------------------------------------------------
C  **        CHECK VIBRATIONS              **
C-----------------------------------------------------------------
  300 CALL VBRCK2(IPER(ICR),INDEXV,KKODE1,ICODE1)
      GO TO 150
  220 DOPTB(ICR)=DELTOB*EFACTB*WEIGB(ICR)**0.5
      RATIO = BMIN/DOPTB(ICR)
      IF (RATIO.GT.0.7) GO TO 150
      ALELOC = 10.0*DHOLE/12.0
      IF(MSYSUN.EQ.1) ALELOC=10.0*DHOLE/100.0
      IF(ALENGT.GT.ALELOC) GO TO 110
      WEIGHT(ICR) = (BMIN/(DELTO*EFACTR))**3.0
  110 IF(RATIO.GT.0.4) GO TO 150
      WEIGHT(ICR) = 0.0
  150 RETURN
      END
      SUBROUTINE DISTPL(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X,Y,Z,D,IBUG)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PURPOSE :                                                           C
C              TO CALCULATE THE DISTANCE FROM A POINT TO A PLANE       C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES  :
C  ***  INPUT  ***
C     X1,Y1,Z1 : THE X, Y, Z COORDINATES OF ONE POINT OF THE PLANE
C     X2,Y2,Z2 : THE X, Y, Z COORDINATES OF A SECOND POINT OF THE PLANE
C     X3,Y3,Z3 : THE X, Y, Z COORDINATES OF A THIRD POINT OF THE PLANE
C     X, Y, Z  : THE POINT OUTSIDE OF THE PLANE
C  ***  OUTPUT  ***
C     D        : THE DISTANCE  OF THE POINT FROM THE PLANE
      IBUG = 0
      A1 = (X2-X1)*(Z3-Z1) - (Z2-Z1)*(Y3-Y1)
      B1 = (Z2-Z1)*(X3-X1) - (X2-X1)*(Z3-Z1)
      C1 = (X2-X1)*(Y3-Y1) - (Y2-Y1)*(X3-X1)
      D1 = Z1*(Y2-Y1)*(X3-X1) + X1*(Z2-Z1)*(Y3-Y1) +
     X     Y1*(X2-X1)*(Z3-Z1) - X1*(Y2-Y1)*(Z3-Z1) -
```

```
     X     Y1*(Z2-Z1)*(X3-X1) - Z1*(X2-X1)*(Y3-Y1)
      A2 = ABS(A1)
      B2 = ABS(B1)
      C2 = ABS(C1)
      DENOM = A2**2.0 + B2**2.0 + C2**2.0
      IF(DENOM.LE.1E-15) GO TO 10
      DD= (A1*X + B1*Y + C1*Z + D1) / (A2**2.0 + B2**2.0 + C2**2.0)**0.5
      D = ABS (DD)
      RETURN
   10 IBUG = 1
      RETURN
      END
      SUBROUTINE DISTPT(X1,Y1,Z1,X,Y,Z,D)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PURPOSE :                                                          C
C             TO CALCULATE THE DISTANCE BETWEEN TWO POINTS.           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  VARIABLES  :
C
C    ***  INPUT  ***
C   X,Y,Z,X1,Y1,Z1 : THE X, Y, Z COORDINATES OF THE TWO POINTS.
C
C    *** OUTPUT  ***
C    D : THE DISTANCE
C
      A = ABS(X-X1)
      B = ABS(Y-Y1)
      C = ABS(Z-Z1)
      D = (A**2.0 + B**2.0 + C**2.0)**0.5
      RETURN
      END
      SUBROUTINE DISPLN(X1,Y1,X2,Y2,X,Y,DISTA,IBUG)
      IBUG = 0
      A1= Y1 - Y2
      B1= X2 - X1
      C1= X1*Y2-X2*Y1
      A = ABS(A1)
      B = ABS(B1)
      DENOM = A**2.0 + B**2.0
      IF(DENOM.LE.1.E-25) GO TO 10
      DISTA=ABS(A1*X + B1*Y + C1)/DENOM**0.5
      RETURN
   10 IBUG = 1
      RETURN
      END
      SUBROUTINE READER(NPFF)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C   PURPOSE : TO READ GENERAL DATA FOR THE DESIGN OF THE VCR BLAST    C
C   THE DATA ARE READ FROM THE FOLLOWING UNITS  :                     C
C   UNIT  7 - TERMINAL ( INTERACTIVELY )                              C
C   UNIT  8 - DISK FILE WITH THE DATA FOR THE CRATERING CURVES        C
C   UNIT  9 - DISK FILE WITH DATA FOR THE STOPE                       C
C   UNIT 10 - INITIATION SEQUENCE
```

```
C   UNIT 11 - DISK FILE WITH DATA FOR THE VIBRATION CHECKS. THIS FILE  CLLM11560
C             IS CREATED BY THE COMPUTER IN A PREVIOUS RUN.            CLLM11570
C                                                                      CLLM11580
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLLM11590
C                                                                      LLM11600
C    VARIABLES :                                                       LLM11610
C          DCONCB : DEPTH OF CONTAINED CHARGE (CYLINDRICAL)            LLM11620
C          DCONCH : DEPTH OF CONTAINED CHARGE (SPHERICAL)              LLM11630
C          DDOUPD : CHARGE LOCATION IN THE BOREHOLE                    LLM11640
C          DELTA  : DEPTH RATIO (SPHERICAL CHARGES)                    LLM11650
C          DELTAB : DEPTH RATIO (CYLINDRICAL CHARGES)                  LLM11660
C          DELTO : OPTIMUM DEPTH RATIO ( SPHERICAL )                   LLM11670
C          DELTOB : OPTIMUM DEPTH RATIO ( CYLINDRICAL )                LLM11680
C          DHOLE : DIAMETER OF BOREHOLE                                LLM11690
C          EXPTYB : EXPLOSIVE TYPE                                     LLM11700
C          EXPTYP : EXPLOSIVE TYPE                                     LLM11710
C          MSYSUN  : 1 METRIC                                          LLM11720
C                    2 IMPERIAL                                        LLM11730
C          NPB1    : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH   LLM11740
C                    RATIO CURVE (CYLINDRICAL)                         LLM11750
C          NPB2    : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH  LLM11760
C                    RATIO CURVE (CYLINDRICAL)                         LLM11770
C          NPFF    : NUMBER OF POINTS FOR THE GENERATION OF THE FREE   LLM11780
C                    FACE                                              LLM11790
C          NP1     : NUMBER OF POINTS IN SCALED CRATER DEPTH - DEPTH   LLM11800
C                    RATIO CURVE (SPHERICAL)                           LLM11810
C          NP2     : NUMBER OF POINTS IN SCALED CRATER RADIUS - DEPTH  LLM11820
C                    RATIO CURVE (SPHERICAL)                           LLM11830
C                                                                      LLM11840
C          NUMBH   :  NUMBER OF HOLES IN THE BLAST                     LLM11850
C          RHOE    : DENSITY OF THE EXPLOSIVE                          LLM11860
C          SCDEPB : SCALED DEPTH OF BURIAL FOR CYLINDRICAL CHARGES     LLM11870
C          SCDEPT : SCALED DEPTH OF BURIAL FOR SPHERICAL CHARGES       LLM11880
C          SCRAD : SCALED RADIUS FOR SPHERICAL                         LLM11890
C          SCRADB : SCALED RADIUS FOR CYLINDRICAL                      LLM11900
C          SCROP : OPTIMUM SCALED RADIUS                               LLM11910
C          SEPDIS : SEPARATION DISTANCE BETWEEN DECKS                  LLM11920
C          TITLE : TITLE                                               LLM11930
C          WSTR  : WEIGHT STRENGTH                                     LLM11940
C          WTCHRB : WEIGHT OF CHARGE IN TESTS (CYLINDRICAL)            LLM11950
C          WTCHRG : WEIGHT OF CHARGE IN TESTS (SPHERICAL)              LLM11960
C          WUPD(I,J)    : WEIGHT OF A DECK I, IN LEVEL J               LLM11970
C          X(I) : X COORDINATE OF HOLE I                               LLM11980
C          XMAX : MAX LOCAL LENGTH (USED TO DETERMINE LENGTH OF BLOCK) LLM11990
C          XMIN : MIN LOCAL LENGTH                                     LLM12000
C          Y(I) : Y COORDINATE OF HOLE I                               LLM12010
C          YMAX : MAX LOCAL WIDTH (USED TO DETERMINE LENGTH OF BLOCK)  LLM12020
C          YYY(I,J) : Y COORDINATE OF CHARGE I IN LEVEL J              LLM12030
C          Z(I) : Z COORDINATE OF HOLE I                               LLM12040
C                                                                      LLM12050
C                                                                      LLM12060
C                                                                      LLM12070
      CHARACTER*1 KUBE, KUBE1                                          LLM12080
      CHARACTER*15 EXPTYP,EXPTYB                                       LLM12090
      CHARACTER*80 TITLE                                               LLM12100
```

```
      CHARACTER*1 IWR,BRTHR                                             LLM12110
      CHARACTER*4 FT,INCH,LBS                                           LLM12120
C-----------------------------------------------------------------------LLM12130
C   **         INCLUDE COMMONS          **                              LLM12140
C-----------------------------------------------------------------------LLM12150
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),        LLM12160
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)   LLM12170
     X,PPVALL(10).AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10).  LLM12180
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,    LLM12190
     XXMAX,XMIN,YMAX.YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),  LLM12200
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS        LLM12210
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTRLLM12220
     X ,NTOTAL,MSYSUN                                                   LLM12230
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS                        LLM12240
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),    LLM12250
     XSCROP,DOPTB(50)                                                   LLM12260
      COMMON/TYPE/EXPTYP,EXPTYB                                         LLM12270
      COMMON/TT/TITLE                                                   LLM12280
      COMMON/FREEF/ XFF(10),YFF(10)                                     LLM12290
      COMMON/ALPHA/ IWR(100,100),BRTHR                                  LLM12300
      COMMON/CONTR/ XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)         LLM12310
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,          LLM12320
     X STDEPT                                                           LLM12330
      COMMON/UNALP/LBS,INCH,FT                                          LLM12340
C-----------------------------------------------------------------------LLM12350
C   **  READ CRATERING CHARACTERISTICS   **                             LLM12360
C-----------------------------------------------------------------------LLM12370
      WRITE(7,8)                                                        LLM12380
    8 FORMAT('  WRITE TITLE TO IDENTIFY PROBLEM')                       LLM12390
      READ (7,1007) TITLE                                               LLM12400
 1007 FORMAT(A80)                                                       LLM12410
      WRITE (7,9)                                                       LLM12420
    9 FORMAT('  ENTER TYPE OF EXPLOSIVE USED FOR TESTS')                LLM12430
      READ (7,1009) EXPTYP                                              LLM12440
 1009 FORMAT(A15)                                                       LLM12450
      WRITE (7,1010)                                                    LLM12460
 1010 FORMAT(' ENTER SYSTEM OF UNITS YOU ARE USING :',/                 LLM12470
     X'      (1) M E T R I C',/                                         LLM12480
     X'      (2) I M P E R I A L ')                                     LLM12490
      READ(7,*) MSYSUN                                                  LLM12500
      IF (MSYSUN.EQ.1) GO TO 1020                                       LLM12510
      LBS=' LBS'                                                        LLM12520
      FT = ' FT '                                                       LLM12530
      INCH ='INCH'                                                      LLM12540
      GO TO 1030                                                        LLM12550
 1020 LBS = ' KG '                                                      LLM12560
      FT = ' M  '                                                       LLM12570
      INCH = ' CM '                                                     LLM12580
 1030 WRITE (7,10) FT,LBS                                               LLM12590
   10 FORMAT ('  ENTER DEPTH OF CONTAINED CHARGE (',A4,'),',/           LLM12600
     X          '          WEIGHT OF EXPLOSIVE (',A4,')',/             LLM12610
     X          '          FOR SPHERICAL CHARGES.',/                   LLM12620
     X          '              --------- ')                            LLM12630
      READ (7,*) DCONCH, WTCHRG                                         LLM12640
      WRITE (7,2010) FT,LBS                                             LLM12650
```

```
 2010 FORMAT ('  ENTER DEPTH OF CONTAINED CHARGE (',A4,'),',/
     X        '          WEIGHT OF EXPLOSIVE (',A4,')',/
     X        '  FOR CYLINDRICAL CHARGES.',/
     X        '          ------------')
      READ (7,*) DCONCB, WTCHRB
      WRITE (7,11)
   11 FORMAT ('  WHAT IS THE WEIGHT STRENGTH OF THE EXPLOSIVE',/
     X        '  RELATIVE TO AN/FO ?')
      READ (7,*) WSTRTH
C-----------------------------------------------------------------------
C  **        READ CRATERING CURVES          **
C-----------------------------------------------------------------------
      WRITE (7,12)
   12 FORMAT ('  DO YOU WANT TO INPUT NEW CRATERING CURVES  ? ')
      READ (7,13) KODE
   13 FORMAT (1A1)
      IF (KODE.EQ.'N') GO TO 30
      WRITE (7,14)
   14 FORMAT ('  ENTER NUMBER OF POINTS TO DESIGN THE ',/
     X        '          SCALED CRATER DEPTH - DEPTH RATIO',/
     X        '          CURVE FOR SPHERICAL CHARGES: ',/
     X        '                    ----------')
      READ (7,*) NP1
      WRITE (8,15) NP1
   15 FORMAT (I5)
      WRITE (7,16)
   16 FORMAT('  ENTER SCALED DEPTH, DEPTH RATIO ',/
     X        '          FOR EACH POINT OF THE CURVE.')
      DO 19 I = 1, NP1
      READ (7,*) SCDEPT(I), DELTA(I)
      WRITE(8,18) SCDEPT(I), DELTA(I)
   18 FORMAT(2F10.4)
   19 CONTINUE
      WRITE(7,20)
   20 FORMAT('  ENTER NUMBER OF POINTS TO DESIGN ',/
     X        '          THE SCALED CRATER RADIUS - DEPTH RATIO',/
     X        '          CURVE FOR SPHERICAL CHARGES :',/
     X        '                    -------- ')
      READ (7,*) NP2
      WRITE(8,15) NP2
      WRITE(7,26)
   26 FORMAT('  ENTER SCALED CRATER RADIUS, DEPTH RATIO',/
     X        '          FOR EACH POINT OF THE CURVE.')
      DO 29 I = 1, NP2
      READ (7,*) SCRAD(I), DELTAR(I)
      WRITE(8,18) SCRAD(I), DELTAR(I)
   29 CONTINUE
      WRITE (7,2014)
 2014 FORMAT ('  ENTER NUMBER OF POINTS TO DESIGN THE ',/
     X        '          SCALED CRATER DEPTH - DEPTH RATIO',/
     X        '          CURVE FOR CYLINDRICAL CHARGES: ',/
     X        '                    ------------')
      READ (7,*) NPB1
      WRITE (8,2015) NPB1
 2015 FORMAT (I5)
```

```
      WRITE (7,2016)                                                    LLM13210
 2016 FORMAT('  ENTER SCALED DEPTH, DEPTH RATIO ',/                     LLM13220
     X        '            FOR EACH POINT OF THE CURVE.')               LLM13230
      DO 2019 I = 1, NPB1                                               LLM13240
      READ (7,*) SCDEPB(I), DELTAB(I)                                   LLM13250
      WRITE(8,2018) SCDEPB(I), DELTAB(I)                                LLM13260
 2018 FORMAT(2F10.4)                                                    LLM13270
 2019 CONTINUE                                                          LLM13280
      WRITE(7,2020)                                                     LLM13290
 2020 FORMAT('  ENTER NUMBER OF POINTS TO DESIGN ',/                    LLM13300
     X        '            THE SCALED CRATER RADIUS - DEPTH RATIO',/    LLM13310
     X        '            CURVE FOR CYLINDRICAL CHARGES :',/           LLM13320
     X        '                          -----------')                  LLM13330
      READ (7,*) NPB2                                                   LLM13340
      WRITE(8,2015) NPB2                                                LLM13350
      WRITE(7,2026)                                                     LLM13360
 2026 FORMAT('  ENTER SCALED CRATER RADIUS, DEPTH RATIO',/              LLM13370
     X        '            FOR EACH POINT OF THE CURVE.')               LLM13380
      DO 2029 I = 1, NPB2                                               LLM13390
      READ (7,*) SCRADB(I), DELTBR(I)                                   LLM13400
      WRITE(8,2018) SCRADB(I), DELTBR(I)                                LLM13410
 2029 CONTINUE                                                          LLM13420
      GO TO 40                                                          LLM13430
C-----------------------------------------------------------------------LLM13440
C  **  READ CURVES FROM FILE  **                                        LLM13450
C-----------------------------------------------------------------------LLM13460
   30 READ (8,15) NP1                                                   LLM13470
      READ (8,18) ((SCDEPT(I), DELTA(I)), I=1,NP1)                      LLM13480
      READ (8,15) NP2                                                   LLM13490
      READ (8,18) ((SCRAD(I), DELTAR(I)), I=1,NP2)                      LLM13500
      READ (8,2015) NPB1                                                LLM13510
      READ (8,2018) ((SCDEPB(I), DELTAB(I)), I=1,NPB1)                  LLM13520
      READ (8,2015) NPB2                                                LLM13530
      READ (8,2018) ((SCRADB(I), DELTBR(I)), I=1,NPB2)                  LLM13540
   40 CONTINUE                                                          LLM13550
C-----------------------------------------------------------------------LLM13560
C   **   READ DATA OF THE STOPE  **                                    LLM13570
C-----------------------------------------------------------------------LLM13580
      WRITE(7,201)                                                      LLM13590
  201 FORMAT ('  ENTER THE DIAMETER OF THE BLASTHOLES (INCHES) :')      LLM13600
      READ (7,*) DHOLE                                                  LLM13610
      WRITE(7,202)                                                      LLM13620
  202 FORMAT('  ENTER NUMBER OF HOLES IN THE BLAST :')                  LLM13630
      READ (7,*) NUMBH                                                  LLM13640
      WRITE(7,204)                                                      LLM13650
  204 FORMAT('  ENTER NUMBER OF ROWS IN THE BLAST :')                   LLM13660
      READ (7,*) NR                                                     LLM13670
      DO 208 I = 1, NR                                                  LLM13680
      WRITE(7,206) I                                                    LLM13690
  206 FORMAT('  ENTER NUMBER OF HOLES IN ROW  ',I2,' :')                LLM13700
      READ (7,*) NCH(I)                                                 LLM13710
  208 CONTINUE                                                          LLM13720
      WRITE(7,2001)                                                     LLM13730
 2001 FORMAT('  ARE THE HOLES BREAKTHROUGH HOLES ? ')                   LLM13740
      READ ( 7,2002) BRTHR                                              LLM13750
```

```
 2002 FORMAT(1A1)                                                       LLM137
C-----------------------------------------------------------------------LLM137
C  **    READ DATA FOR VIBRATION CHECKS        **                       LLM137
C     SPECIFY POINTS OF PARTICULAR INTEREST AND ACCEPTABLE              LLM137
C     VIBRATIONS IN THESE POINTS.                                       LLM138
C-----------------------------------------------------------------------LLM138
      WRITE (7,250)                                                     LLM138
  250 FORMAT('     THE FOLLOWING DATA ARE FOR CHECKING VIBRATION',/     LLM138
     X        '     LEVELS AT PILLARS  AND OTHER PARTICULAR POINTS',/   LLM138
     X        '     OF INTEREST ....',/                                 LLM138
     X        '     ARE THESE NEW OR PREVIOUS DATA ?')                  LLM138
      READ (7,13) KODE                                                  LLM138
      IF (KODE.EQ.'P') GO TO 490                                        LLM138
      WRITE(7,270)                                                      LLM138
  270 FORMAT('    ENTER THE NUMBER OF POINTS IN WHICH VIBRATIONS ARE',/ LLM139
     X        '    TO BE CHECKED:')                                    LLM139
      READ (7,*) NCHK                                                   LLM139
      IF(NCHK.EQ.0) GO TO  331                                          LLM139
      WRITE(11,280)                                                     LLM139
  280 FORMAT(I5)                                                        LLM139
      WRITE(7,300)                                                      LLM139
  300 FORMAT('    ENTER SPECIFIC POINTS IN WHICH VIBRATIONS WILL BE',   LLM139
     X        ' CHECKED.',/                                             LLM139
     X'  ( EACH POINT MUST BE SPECIFIED BY ITS COORDINATES SEPARATED',/ LLM139
     X'   BY COMMAS, AND THE MAXIMUM ACCEPTABLE LEVEL OF PEAK PARTICLE' LLM140
     X,/'     VELOCITY. E.G. 12.4,12.6,14.2,10.0)')                    LLM140
      DO 315 I = 1, NCHK                                                LLM140
      READ (7,*) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)                LLM140
C-----------------------------------------------------------------------LLM140
C  **  CONVERT TO LOCAL COORDINATES    **                               LLM140
C-----------------------------------------------------------------------LLM140
      REWIND 51                                                         LLM140
      READ(51,519) TITLE                                                LLM140
      READ(51,520) XREFER,YREFER,ZREFER                                 LLM140
      READ(51,520) STLENG,STWIDT,STDEPT                                 LLM141
      READ(51,520) STANGR                                               LLM141
      STANGR=3.14159*STANGR/180.0                                       LLM141
  519 FORMAT(A80)                                                       LLM141
  520 FORMAT(4F10.2)                                                    LLM141
      XVCHK(I) = XVCHK(I) - XREFER                                      LLM141
      YVCHK(I) = YVCHK(I) - YREFER                                      LLM141
      ZVCHK(I) = ZVCHK(I) - ZREFER                                      LLM141
      XC = XVCHK(I)*COS(STANGR) + YVCHK(I)*SIN(STANGR)                  LLM141
      YC = -XVCHK(I)*SIN(STANGR) + YVCHK(I)*COS(STANGR)                 LLM141
      XVCHK(I) = XC                                                     LLM142
      YVCHK(I) = YC                                                     LLM142
      WRITE(11,310) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)             LLM142
  310 FORMAT(4F15.5)                                                    LLM142
  315 CONTINUE                                                          LLM142
      WRITE(7,320)                                                      LLM142
  320 FORMAT('  ENTER THE SITE FACTORS K, M FOR THE CALCULATION OF THE',LLM142
     X' PEAK PARTICLE',/'                VELOCITY :')                   LLM142
      READ (7,*) AKFAC, AMFAC                                           LLM142
      WRITE(11,330) AKFAC, AMFAC                                        LLM142
  330 FORMAT (2F15.5)                                                   LLM143
```

```
  331 CONTINUE                                                          LLM14310
C-----------------------------------------------------------------------LLM14320
C  **        ENTER DATA FOR PILLARS        **                           LLM14330
C-----------------------------------------------------------------------LLM14340
      WRITE (7,430)                                                     LLM14350
  430 FORMAT('  ARE THERE ANY PILLARS IN WHICH YOU WOULD LIKE TO CHECK 'LLM14360
     X,'VIBRATIONS ?')                                                  LLM14370
      READ (7,431) KODE1                                                LLM14380
  431 FORMAT (1A1)                                                      LLM14390
      IF (KODE1.EQ.'N') GO TO 490                                       LLM14400
      WRITE(7,440)                                                      LLM14410
  440 FORMAT('  ENTER NUBER OF PILLARS :')                              LLM14420
      READ (7,*) NUMPIL                                                 LLM14430
      WRITE(11,450) NUMPIL                                              LLM14440
  450 FORMAT(I5)                                                        LLM14450
      DO 480 I = 1, NUMPIL                                              LLM14460
      WRITE(7,460) I                                                    LLM14470
  460 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 1ST POINT OF PILLAR', LLM14480
     X' NO.',I3)                                                        LLM14490
      READ (7,*) XPIL1(I), YPIL1(I), ZPIL1(I)                           LLM14500
      WRITE(11,470) XPIL1(I), YPIL1(I), ZPIL1(I)                        LLM14510
  470 FORMAT (3F15.5)                                                   LLM14520
      WRITE(7,471) I                                                    LLM14530
  471 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 2ND POINT OF PILLAR', LLM14540
     X' NO.', I3)                                                       LLM14550
      READ (7,*) XPIL2(I), YPIL2(I), YPIL2(I)                           LLM14560
      WRITE(11,470) XPIL2(I), YPIL2(I), ZPIL2(I)                        LLM14570
      WRITE(7,473) 1                                                    LLM14580
  473 FORMAT('  ENTER X, Y, Z, COORDINATES OF THE 3RD POINT OF PILLAR', LLM14590
     X' NO.', I3)                                                       LLM14600
      READ (7,*) XPIL3(I), YPIL3(I), ZPIL3(I)                           LLM14610
      WRITE(11,470) XPIL3(I), YPIL3(I), ZPIL3(I)                        LLM14620
  480 CONTINUE                                                          LLM14630
      WRITE(7,482)                                                      LLM14640
  482 FORMAT ('  ENTER MAXIMUM ACCEPTABLE LEVEL FOR PEAK PARTICLE ',    LLM14650
     X 'VELOCITY ',/'  IN THE PILLAR')                                  LLM14660
      READ (7,*) PPVPAL                                                 LLM14670
      WRITE(11,484) PPVPAL                                              LLM14680
  484 FORMAT(I5)                                                        LLM14690
      GO TO 500                                                         LLM14700
C-----------------------------------------------------------------------LLM14710
C    **        READ FROM FILE        **                                 LLM14720
C-----------------------------------------------------------------------LLM14730
  490 IF (KODE.NE.'P') GO TO 500                                        LLM14740
      READ (11,280) NCHK                                                LLM14750
      DO 420 I = 1, NCHK                                                LLM14760
      READ (11,310) XVCHK(I), YVCHK(I), ZVCHK(I), PPVALL(I)             LLM14770
  420 CONTINUE                                                          LLM14780
      READ(11,450) NUMPIL                                               LLM14790
      DO 498 I = 1, NUMPIL                                              LLM14800
      READ(11,470) XPIL1(I),YPIL1(I),ZPIL1(I)                           LLM14810
      READ(11,470) XPIL2(I),YPIL2(I),ZPIL2(I)                           LLM14820
      READ(11,470) XPIL3(I),YPIL3(I),ZPIL3(I)                           LLM14830
  498 CONTINUE                                                          LLM14840
      READ(11,484) PPVAL                                                LLM14850
```

```
  500 CONTINUE
C-----------------------------------------------------------------------
C  **   READ COORDINATES OF THE FREE FACE    **
C-----------------------------------------------------------------------
      READ (9,*) NPFF
      READ (9,213)((XFF(I),YFF(I)),I=1,NPFF)
  213 FORMAT(2F15.5)
C-----------------------------------------------------------------------
C  **   CONVERT TO LOCAL COORDINATES    **
C-----------------------------------------------------------------------
      DO 511 I = 1,NPFF
      XFF(I) = XFF(I) - XREFER
      YFF(I) = YFF(I) - YREFER
      XC = XFF(I)*COS(STANGR) + Y(I)*SIN(STANGR)
      YC = -XFF(I)*SIN(STANGR) + YFF(I)*COS(STANGR)
      XFF(I) = XC
      YFF(I) = YC
  511 CONTINUE
C-----------------------------------------------------------------------
C  **  READ LINE OF FINAL WALL IN THE TOP LEVEL **
C-----------------------------------------------------------------------
      READ(12,*) NCONTR
      IF (NCONTR.EQ.0) GO TO 4100
      DO 4050 I = 1,NCONTR
      READ(12,4051) XCONT1(I),YCONT1(I),XCONT2(I),YCONT2(I)
 4051 FORMAT(4F10.2)
      XCONT1(I) = XCONT1(I) - XREFER
      YCONT1(I) = YCONT1(I) - YREFER
      XC = XCONT1(I)*COS(STANGR) + YCONT1(I)*SIN(STANGR)
      YC = -XCONT1(I)*SIN(STANGR) + YCONT1(I)*COS(STANGR)
      XCONT1(I) = XC
      YCONT1(I) = YC
      XCONT2(I) = XCONT2(I) - XREFER
      YCONT2(I) = YCONT2(I) - YREFER
      XC = XCONT2(I)*COS(STANGR) + YCONT2(I)*SIN(STANGR)
      YC = -XCONT2(I)*SIN(STANGR) + YCONT2(I)*COS(STANGR)
      XCONT2(I) = XC
      YCONT2(I) = YC
 4050 CONTINUE
 4100 CONTINUE
C-----------------------------------------------------------------------
C   **      ENTER DATA FOR THE EXPLOSIVE WHICH IS USED       **
C-----------------------------------------------------------------------
      WRITE(7,509)
  509 FORMAT('  ENTER TYPE OF EXPLOSIVE TO BE USED :')
      READ(7,1509) EXPTYB
 1509 FORMAT(A15)
      WRITE(7,507)
  507 FORMAT('  WHAT IS THE WEIGHT STRENGTH OF THE  EXPLOSIVE YOU ARE',
     X' USING  ?')
      READ (7,*) WSTR
C-----------------------------------------------------------------------
      WRITE(7,508) LBS,INCH
  508 FORMAT('  WHAT IS THE DENSITY OF THE EXPLOSIVE YOU ARE USING ?',/
     X'     ',A4,'/ CUBIC ',A4,')')
```

```
      READ (7,*) RHOE
      WRITE (7,528)
  528 FORMAT(' ENTER STOPE LENGTH, STOPE WIDTH AND STOPE DEPTH :')
      READ (7,*) XMAX, YMAX, ZMAX
      XMIN = 0.0
      YMIN = 0.0
      ZMIN = 0.0
      ELEV = ZMAX
      CALL PLVIEW(ELEV,X,Y,NUMBH)
      DO 8000 I = 1,NUMBH
      Z(I) = ZMAX
 8000 CONTINUE
      RETURN
      END
      SUBROUTINE INITB(ICODE,NCALL,IA,IFIRST,KKODEI)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   PURPOSE  :   TO PROVIDE THE INITIATION SEQUENCE IN A TYPICAL      C
C                UNDERGROUND BENCH.                                    C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC C
C
C   VARIAPLES :
C        REFER TO MAIN PROGRAM
C
C
      CHARACTER*1 KKODEI,IWR,BRTHR
      CHARACTER*80 TITLE
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTR
     X , NTOTAL
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)
      COMMON/DXDY/ DX,DY
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB
      COMMON/XBLK/XB(100),YB(100)
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NOECKS
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),
     XSCROP,DOPTB(50)
      COMMON/TYPE/EXPTYP.EXPTYB
      COMMON/TT/TITLE
      COMMON/FREEE/ XFF(10),YFF(10)
      COMMON/ALPHA/IWR(100,100),BRTHR
      COMMON/CONTR/XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,
     X STDEPT
      IF (NCALL.EQ.2) GO TO 500
C-----------------------------------------------------------------------
C  **      INITIALIZE VARIABLES      **
C-----------------------------------------------------------------------
      ICODE = 0
```

```
      IVBEF = 0                                                         LLM159
      IBEF  = 0                                                         LLM159
      INDEXV = 0                                                        LLM159
      IF(KKODE1.EQ.'M') GO TO 500                                       LLM159
      WRITE (7,17)                                                      LLM160
   17 FORMAT('      AUTOMATIC GENERATION OF THE INITIATION SEQUENCE',/  LLM160
     X        '      ----------------------------------------------')   LLM160
C-----------------------------------------------------------------------LLM160
C     TAKE FIRST ROW AND FIND CHARGE IN THE CENTER OF THE ROW. THIS     LLM160
C     CHARGE WILL DETONATE FIRST.                                       LLM160
C-----------------------------------------------------------------------LLM160
      III = NCH(1)/2                                                    LLM160
      KP = NCH(1)                                                       LLM160
      DO 20 I=1,KP                                                      LLM160
      IPER(I) = 1 + ABS(III-1)                                          LLM161
   20 CONTINUE                                                          LLM161
C-----------------------------------------------------------------------LLM161
C  **    SUBSEQUENT ROWS    **                                          LLM161
C-----------------------------------------------------------------------LLM161
      DO 50 LR = 2, NR                                                  LLM161
      I1 = NCH(LR) / 2                                                  LLM161
      KPP = NCH(LR)                                                     LLM161
      DO 40 I = 1, KPP                                                  LLM161
      IPER( KP+ I) = IPER(III) + ABS(I1-I) + 2                          LLM161
   40 CONTINUE                                                          LLM162
      III = I1 + KP                                                     LLM162
      KP = KP + KPP                                                     LLM162
   50 CONTINUE                                                          LLM162
C-----------------------------------------------------------------------LLM162
C  **     FIND MAXIMUM PERIOD    **                                    LLM162
C-----------------------------------------------------------------------LLM162
      IA = 0                                                            LLM162
      DO 60 I = 1, NUMBH                                                LLM162
      IF(IPER(I).GE.100) GO TO 60                                       LLM162
      IF(IA.GT.IPER(I)) GO TO 60                                        LLM163
      IA = IPER(I)                                                      LLM163
   60 CONTINUE                                                          LLM163
C-----------------------------------------------------------------------LLM163
C  **  CHECK VIBRATIONS FOR EACH DELAY PERIOD  **                       LLM163
C-----------------------------------------------------------------------LLM163
   65 DO 70  IV = 1,IA                                                  LLM163
      CALL VBRCHK(IV,INDEXV,KKODE1,ICODE1 )                             LLM163
      IF (INDEXV.EQ.1) GO TO 80                                         LLM163
   70 CONTINUE                                                          LLM163
      DO 71 IKOP=1,NUMBH                                                LLM164
   71 CONTINUE                                                          LLM164
      RETURN                                                            LLM164
C-----------------------------------------------------------------------LLM164
C   VIBRATIONS EXCEED MAXIMUM ACCEPTABLE. FIND LAST CHARGE WITH PERIOD  LLM164
C   EQUAL TO IV.                                                        LLM164
C-----------------------------------------------------------------------LLM164
   80 DO 90 I = 1, NUMBH                                                LLM164
      IF ( IPER(I).LT.IV) GO TO 90                                      LLM164
      ICOUNT = I                                                        LLM164
   90 CONTINUE                                                          LLM165
```

```
      IF (IV.NE.IVBEF) IBEF = 100000
C-----------------------------------------------------------------------
C  **  PERIODS FROM CHARGE ICOUNT ON ARE INCREASED BY ONE  **
C-----------------------------------------------------------------------
      DO 100 I = 1, NUMBH
      IF ((IPER(I).GE.IV).AND.(I.GE.ICOUNT)) GO TO 120
      GO TO 100
  120 IF (I.GE.IBEF) GO TO 130
      IPER(I) = IPER(I) + 1
  100 CONTINUE
  130 IVBEF = IV
      IBEF = ICOUNT
      GO TO 65
  500 ICODE = 1
      IFIRST = IFIRST + 1
  505 WRITE(7,501) IFIRST
  501 FORMAT('  HOW MANY CHARGES ARE DETONATED IN PERIOD # ',I2,' ?')
      READ (7,*) NCD1
      WRITE(7,510) IFIRST
  510 FORMAT('  ENTER NUMBERS INDICATING CHARGES DETONATING IN PERIOD'.
     X' # ',I2,'  :')
      DO 520 I = 1,NCD1
      READ(7,*) IDET
      IPER(IDET) = IFIRST
  520 CONTINUE
      CALL VBRCHK(IFIRST,INDEXV,KKODE1,ICODE1)
      IF(INDEXV.EQ.1) GO TO 505
      RETURN
      END
      SUBROUTINE SEQOUT(K,ISOUT,ICODE)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C   PURPOSE  :                                                          C
C                TO DETERMINE THE FINAL SEQUENCE OF AN UNDERGROUND      C
C                BENCH BLAST.                                           C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES : REFER TO MAIN PROGRAM
C
C
C
      INTEGER*4 CLTIME,TCPU
      CHARACTER*15 EXPTYP,EXPTYR
      CHARACTER*80 TITLE
      CHARACTER*8  QDATE ,QTIM
      CHARACTER*1 KODE,IWR,BRTHR,KKODE1,KKODEI
      COMMON /RDRDT1/ QCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR.
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,QCONCB,SCDEPB(50).
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTR
     X ,NTOTAL,MSYSUN
```

```
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)                       LLM17060
      COMMON/DXDY/ DX,DY                                                LLM17070
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB                            LLM17080
      COMMON/XBLK/XB(100),YB(100)                                       LLM17090
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS                        LLM17100
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),    LLM17110
     XSCROP,DOPTB(50)                                                   LLM17120
      COMMON/TYPE/EXPTYP,EXPTYB                                         LLM17130
      COMMON/TT/TITLE                                                   LLM17140
      COMMON/FREEF/ XFF(10),YFF(10)                                     LLM17150
      COMMON/UPDATE/XXX(50,20),YYY(50,20),ZZZ(50,20),WUPD(50,20)        LLM17160
      COMMON/ALPHA/IWR(100,100),BRTHR                                   LLM17170
      COMMON/CONTR/ XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)         LLM17180
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,          LLM17190
     X STDEPT                                                           LLM17200
      DIMENSION ISOUT(50,20)                                            LLM17210
C-----------------------------------------------------------------------LLM17220
C  **     INITIALIZE VARIABLES     **                                   LLM17230
C-----------------------------------------------------------------------LLM17240
      INDEXV = 0                                                        LLM17250
      IVREF = 100000                                                    LLM17260
C-----------------------------------------------------------------------LLM17270
C  **    GET THE FIRST LEVEL OF CHARGES  ( DECKS )     **               LLM17280
C-----------------------------------------------------------------------LLM17290
      DO 20 I = 1,NUMBH                                                 LLM17300
      ISOUT(I,1) = IPER(I)                                              LLM17310
   20 CONTINUE                                                          LLM17320
      IF ( K.EQ.1 ) RETURN                                              LLM17330
C-----------------------------------------------------------------------LLM17340
C  **   GO TO SUBSEQUENT LEVELS    **                                  LLM17350
C-----------------------------------------------------------------------LLM17360
      DO 50 I = 1, NUMBH                                                LLM17370
      ISOUT(I,K) = ISOUT(1,K-1) + 2                                     LLM17380
   50 CONTINUE                                                          LLM17390
      DO 150 I = 1, NUMBH                                               LLM17400
      IF(ISOUT(I,K-1).GE.1000.0) GO TO 150                              LLM17410
      IF(ISOUT(I,K-1).LT.ISOUT(I,K)) GO TO 150                          LLM17420
      IDIFF = ISOUT(I,K-1)-ISOUT(I,K) + 1                               LLM17430
      DO 100 IS = 1,NUMBH                                               LLM17440
      IF ( (ISOUT(IS,K).GE.ISOUT(I,K)).AND.(IS.GT.I)) ISOUT(IS,K)=      LLM17450
     XISOUT(IS,K)+IDIFF                                                 LLM17460
  100 CONTINUE                                                          LLM17470
  150 CONTINUE                                                          LLM17480
C-----------------------------------------------------------------------LLM17490
C  **   FIND MAXIMUM PERIOD  (SO FAR)   **                              LLM17500
C-----------------------------------------------------------------------LLM17510
  165 IA = 0                                                            LLM17520
      INDEXV = 0                                                        LLM17530
      DO 160 I = 1, NUMBH                                               LLM17540
      IF(ISOUT(I,K).LE.IA) GO TO 160                                    LLM17550
      IF(ISOUT(I,K).GE.1000) GO TO 160                                  LLM17560
      IA = ISOUT(I,K)                                                   LLM17570
  160 CONTINUE                                                          LLM17580
C-----------------------------------------------------------------------LLM17590
C  **   EXAMINE VIBRATIONS FOR EACH PERIOD  **                          LLM17600
```

```
C-----------------------------------------------------------------------
      DO 170 IV = 1, 16
      CALL VIBRM(IV,INDEXV,K,ISOUT)
      IF (INDEXV.EQ.1) GO TO 180
  170 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C  **   VIBRATIONS EXCEED MAXIMUM ACCEPTABLE LEVEL  **
C       FIND LAST CHARGE WITH PERIOD EQUAL TO  IV
C-----------------------------------------------------------------------
  180 DO 190 I = 1, NUMBH
      IF (ISOUT(I,K).NE.IV) GO TO 190
      ICOUNT = I
  190 CONTINUE
      IF(IV.NE.IVBEF) IBEF = 100000
C-----------------------------------------------------------------------
C  ** PERIODS FROM CHARGE ICOUNT ON ARE INCREASED BY ONE **
C-----------------------------------------------------------------------
      DO 200 I = 1, NUMBH
      IF ((ISOUT(I,K).GE.IV).AND.(I.GE.ICOUNT)) GO TO 220
      GO TO 200
  220 IF(I.GE.IBEF) GO TO 230
      ISOUT(I,K) = ISOUT(I,K) + 1
  200 CONTINUE
  230 IVBEF = IV
      IBEF = ICOUNT
      GO TO 165
      END
      SUBROUTINE SEPDST(IDELTA,BURDPF)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C  PURPOSE  :                                                         C
C       TO CALCULATE THE SEPARATON DISTANCE BETWEEN CHARGES IN THE    C
C       BOREHOLE                                                      C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C      REFER TO MAIN PROGRAM
C
      CHARACTER*80 TITLE
      CHARACTER*1 BRTHR,IWR
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTP
     X ,NTOTAL,MSYSUN
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)
      COMMON/DXDY/ DX,DY
      COMMON/DELT/DELTD,EFACTR,DELTDB,EFACTB
      COMMON/XBLK/XB(100),YB(100)
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS
```

```
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50).RCR(50),DEPTH(50),
     XSCROP,DOPTB(50)
      COMMON/TYPE/EXPTYP,EXPTYB
      COMMON/ALPHA/IWR(100.100),BRTHR
      COMMON/TT/TITLE
      COMMON/FREEF/ XFF(10),YFF(10)
      COMMON/CONTR/XCONT1(10).XCONT2(10),YCONT1(10).YCONT2(10)
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR.STLENG.STWIDT,
     X STDEPT
      DIMENSION WMINN(50),ZLEFT(50),CHARGB(50,15)
      PI = 3.14159
      ICHKPT = 0
      TOP = 0
      BOT = 0
      ZMAX = 10000000.0
      DO 5 I = 1,NUMBH
      IF(ZMAX.LT.Z(I)) GO TO 5
      ZMAX = Z(I)
    5 CONTINUE
      DO 6 I = 1, NUMBH
      ZLEFT(I) = Z(I) - ZMAX
    6 CONTINUE
      WMINA = 10000000.0
      WRITE(7,10)
   10 FORMAT('  ENTER POWDER FACTOR FOR THE BLAST :')
      READ(7,*) POWDFC
      WRITE(7,20)
   20 FORMAT('  ENTER BURDEN AND SPACING OF THE PATTERN :')
      READ (7,*) BURDPF,SPACPF
      VOLUME = BURDPF*SPACPF*(ZMAX-ZMIN)
C-----------------------------------------------------------------------
C  **  ENTER FACTOR FOR THE DETERMINATION OF MAXIMUM SEPARATION
C      DISTANCE FOR ACCEPTABLE FRAGMENTATION **
C-----------------------------------------------------------------------
      WRITE(7,25)
   25 FORMAT('  ENTER MAXIMUM DEPTH RATIO FOR ACCEPTABLE ',
     X'FRAGMENTATION :',/'   TYPICAL VALUES :',/'    HARD ROCK : 0.75',
     X/'    SOFT ROCK : 0.85')
      READ(7,*) AMXDRT
C-----------------------------------------------------------------------
C  **   WEIGHT OF EXPLOSIVE PER HOLE    **
C-----------------------------------------------------------------------
      WEXPH = VOLUME * POWDFC
C-----------------------------------------------------------------------
C  **    TOTAL LENGTH OF EXPLOSIVE PER HOLE    **
C-----------------------------------------------------------------------
      ALEXP = WEXPH/((RHOE*DHOLE**2.0*PI/4.0)*12.0)
      IF(MSYSUN.EQ.1)ALEXP=WEXPH/((RHOE/1000.0*DHOLE**2.0*PI/4.0)*100.0)
C-----------------------------------------------------------------------
C  **   TOTAL LENGTH OF STEMMING COLUMN PER HOLE    **
C-----------------------------------------------------------------------
      ALSTEM = ZMAX - ZMIN - ALEXP
       IDELTA = 1
       IF(BRTHR.EQ.'N') IDELTA = 0
C-----------------------------------------------------------------------
```

```
C  **  EXAMINE VIBRATIONS  **                                          LLM18710
C-----------------------------------------------------------------------LLM18720
      NN = 0                                                            LLM18730
      ZM = (ZMAX+ZMIN)/2.0                                              LLM18740
      NN = NCHK + NUMPIL                                                LLM18750
      IF ( NN.EQ.0) GO TO 100                                           LLM18760
      ELEV = ZM                                                         LLM18770
      CALL PLVIEW(ELEV,X,Y,NUMBH)                                       LLM18780
      XM = 0.0                                                          LLM18790
      YM = 0.0                                                          LLM18800
      DO 50 IA=1,NUMBH                                                  LLM18810
      XM = XM + X(IA)                                                   LLM18820
      YM = YM + Y(IA)                                                   LLM18830
   50 CONTINUE                                                          LLM18840
      XM = XM/NUMBH                                                     LLM18850
      YM = YM/NUMBH                                                     LLM18860
      WMIN = 1000000.0                                                  LLM18870
      DO 30 I = 1,NCHK                                                  LLM18880
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),XM,YM,ZM,DIST)             LLM18890
      WEIGMX = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DIST**2.0                 LLM18900
      IF(WEIGMX.LT.WMIN) WMIN = WEIGMX                                  LLM18910
   30 CONTINUE                                                          LLM18920
      IF (WMINA.GT.WMIN) WMINA=WMIN                                     LLM18930
C-----------------------------------------------------------------------LLM18940
C  **    FIND NUMBER OF DECKS      **                                   LLM18950
C-----------------------------------------------------------------------LLM18960
  100 NDECKS = IFIX(WEXPH/WMINA + 0.5)                                  LLM18970
      IF(NDECKS.GT.0) GO TO 120                                         LLM18980
      NDECKS = 1                                                        LLM18990
      ALENGT = ALEXP                                                    LLM19000
      ALST = ALSTEM                                                     LLM19010
      GO TO 200                                                         LLM19020
  120 ALENGT = ALEXP/NDECKS                                             LLM19030
C-----------------------------------------------------------------------LLM19040
C   ** IDELTA = 0 FOR NON BREAKTHROUGH HOLES  **                        LLM19050
C   ** IDELTA = 1 FOR     BREAKTHROUGH HOLES  **                        LLM19060
C-----------------------------------------------------------------------LLM19070
      ALST = ALSTEM/FLOAT(NDECKS + IDELTA)                              LLM19080
C-----------------------------------------------------------------------LLM19090
C  **   OPTIMUM SEPARATION DISTANCE   **                                LLM19100
C-----------------------------------------------------------------------LLM19110
      IF(MSYSUN.EQ.1) GO TO 291                                         LLM19120
  200 SEPM1 = ((8.0*DHOLE*PI*DHOLE**2.0/4.0)*RHOE)**(1.0/3.0)*          LLM19130
     X EFACTRADELTO - 4.0*DHOLE/12.0                                    LLM19140
      GO TO 292                                                         LLM19150
  291 SEPM1 = ((8.0*DHOLE*PI*DHOLE**2.0/4.0)*RHOE*0.001)**(1.0/3.0)*    LLM19160
     X EFACTRADELTO - 4.0*DHOLE/100.0                                   LLM19170
  292 SEPM2 = 1000000.0                                                 LLM19180
      WXXX = 8.0*DHOLE*PI*DHOLE**2.0/4.0*RHOE                           LLM19190
      IF(MSYSUN.EQ.1) WXXX = WXXX/1000.0                                LLM19200
      IF(WMINA.GE.WXXX) GO TO 201                                       LLM19210
      IF(MSYSUN.EQ.1) GO TO 295                                         LLM19220
      SEPM2 = WMINA**(1.0/3.0)*EFACTRADELTO-((WMINA/(RHOE*PI*DHOLE**2.0 LLM19230
     X/4.0))/12.0)/2.0                                                  LLM19240
      GO TO 201                                                         LLM19250
```

```
  295 SEPM2 = WMINA**(1.0/3.0)*EFACTR*DELTO-((WMINA/(RHOE*PI*DHOLE**2.0 LLM19260
     X/(4.0*1000.0)))/100.0)/2.0                                        LLM19270
  201 SEPMAX = MIN(SEPM1,SEPM2)                                         LLM19280
      IF(NN.EQ.0) SEPMAX = SEPM1                                        LLM19290
C-----------------------------------------------------------------------LLM19300
C  **  MINIMUM SEPARATION DISTANCE  (GENERALLY ACCEPTED)  **            LLM19310
C-----------------------------------------------------------------------LLM19320
      SEPMIN = 1.25*DHOLE                                               LLM19330
      IF(MSYSUN.EQ.1) SEPMIN = 1.25*DHOLE/(2.54*3.33)                   LLM19340
      IF (ALST.LE.SEPMAX) GO TO 300                                     LLM19350
  202 NDECKS = ALSTEM /SEPMAX - IDELTA + 0.5                            LLM19360
  205 WXXX =  WEXPH / NDECKS                                            LLM19370
      IF ( WXXX.GT.WMINA ) GO TO 203                                    LLM19380
      IF(NDECKS.GT.0) GO TO 220                                         LLM19390
      ALENGT = ALEXP                                                    LLM19400
      ALST = ALSTEM                                                     LLM19410
      NDECKS = 1                                                        LLM19420
      GO TO 300                                                         LLM19430
  220 ALENGT = ALEXP/FLOAT(NDECKS)                                      LLM19440
      ALST = ALSTEM/FLOAT(NDECKS + IDELTA)                              LLM19450
  300 SEPDIS = ALST                                                     LLM19460
      IF(SEPDIS.GE.SEPMIN) GO TO 350                                    LLM19470
      IF(SEPMAX.GT.SEPMIN) GO TO 301                                    LLM19480
      NDECKS = NDECKS -1                                                LLM19490
      GO TO 205                                                         LLM19500
  301 SEPMAX = SEPMIN                                                   LLM19510
      ICHKPT = 1                                                        LLM19520
      GO TO 202                                                         LLM19530
  203 WRITE(7,308)                                                      LLM19540
  308 FORMAT('  THE SEPARATION DISTANCE BETWEEN DECKS IS UNACCEPTABLY', LLM19550
     X ' SMALL.',/,'  EXECUTION STOPS HERE.')                           LLM19560
      STOP                                                              LLM19570
C-----------------------------------------------------------------------LLM19580
C     CHECK BOTTOM PARTS OF THE HOLES TO CONSIDER THE IRREGULARITIES    LLM19590
C     OF THE INITIAL FACE.                                              LLM19600
C-----------------------------------------------------------------------LLM19610
  350 IF (ICHKPT.EQ.0) GO TO 351                                        LLM19620
      CH1=(8.0*DHOLE*PI*DHOLE**2.0/4.0)*RHOE                            LLM19630
      IF(MSYSUN.EQ.1) CH1 = CH1/1000.0                                  LLM19640
      CH2=WMINA                                                         LLM19650
      CH3=MIN(CH1,CH2)                                                  LLM19660
      DC1=EFACTR*CH3**(1/3.0)                                           LLM19670
      DCR=SEPDIS/DC1                                                    LLM19680
      IF(DCR.LT.AMXDRT) GO TO 351                                       LLM19690
      WRITE(7,307)                                                      LLM19700
      WRITE(6,307)                                                      LLM19710
  307 FORMAT('  THE SEPARATION DISTANCE BETWEEN DECKS IS LARGE.',/      LLM19720
     X '   FRAGMENTATION WILL BE UNACCEPTABLE IN MANY CASES.')          LLM19730
  351 DO 600 I = 1,NUMRH                                                LLM19740
      IF(SEPMAX.GT.(ZLEFT(I)+ALST)) GO TO 600                           LLM19750
      IF((ALST+ALENGT).GT.ZLEFT(I)) GO TO 500                           LLM19760
      CHARGB(I,1)=WMINA                                                 LLM19770
      ZLEFT(I) = ZLEFT(I)-ALENGT                                        LLM19780
      GO TO 600                                                         LLM19790
  500 CHARGB(I,1) = WMINA*ALENGT/(ALST+ALENGT)                          LLM19800
```

```
  600 CONTINUE
      RETURN
      END
      SUBROUTINE DISTLN(X1,Y1,X2,Y2,X3,Y3,DIST)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C    PURPOSE  :                                                      C
C       TO CALCULATE DISTANCE OF ONE POINT TO A CERTAIN LINE         C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C
C     X1.X2,Y1,Y2 : X, Y COORDINATES OF THE LINE
C     X3,Y3       : X, Y COORDINATES OF THE POINT
C
C
      DO 20 I=1,10
      XM = (X1+X2)/2.0
      YM = (Y1+Y2)/2.0
      DM = ((ABS(X3-XM))**2.0+(ABS(Y3-YM))**2.0)**0.5
      D1 = ((ABS(X3-X1))**2.0+(ABS(Y3-Y1))**2.0)**0.5
      D2 = ((ABS(X3-X2))**2.0+(ABS(Y3-Y2))**2.0)**0.5
      IF(D1.GT.DM) GO TO 5
      X2 = XM
      Y2 = YM
      GO TO 20
    5 X1 = XM
      Y1 = YM
   20 CONTINUE
      DIST = ((ABS(X3-XM))**2.0+(ABS(Y3-YM))**2.0)**0.5
      RETURN
      END
      SUBROUTINE TOPH (N,TOP)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C   PURPOSE :                                                        C
C       TO ADJUST CHARGE WEIGHT SO THAT BACKBREAK AT THE TOP IS       C
C       AVOIDED.                                                     C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES : REFER TO MAIN PROGRAM
C
      INTEGER*4 CLTIME,TCPU
      CHARACTER*1 KODE,IWR,ORTHR,KKODE1,KKODE1
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10).
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX.ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50).
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT.BMIN.SEPDIS
      COMMON/RDRDT2/ NP1,NPB1.NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTR
     X ,NTOTAL,MSYSUN
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB
```

```
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),
     XSCROP,DOPTB(50)
      COMMON/CONTR/ XCONT1(10),XCONT2(10),YCONT1(10),YCONT2(10)
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,
     X STDEPT
      PI = 3.14159
      WXX1 = (8.0*DHOLE*PI*DHOLE**2.0/4.0)*RHOE
      IF(MSYSUN.EQ.1) WXX1 = WXX1/1000.0
      TOP = SEPDIS
      WXX2 = WEIGHT(N)
      WXXX = MIN(WXX1,WXX2)
      CHAR2 = (((WXXX/RHOE)/(DHOLE**2.0*PI/4.0))/2.0)/12.0
      IF(MSYNUN.EQ.1)
     XCHAR2 = (((WXXX/(RHOE/1000.0))/(DHOLE**2.0*PI/4.0))/2.0)/100.0
      DELTOT=(SEPDIS+CHAR2)/(EFACTR*WXXX**(1.0/3.0))
      DO 20 I = 2,NP2
      IF(DELTAR(I).GE.DELTOT) GO TO 21
   20 CONTINUE
   21 SCRTOP=SCRAD(I-1)+(DELTOT-DELTAR(I-1))/(DELTAR(I)-DELTAR(I-1))*
     X (SCRAD(I)-SCRAD(I-1))
      DCONTM = 100000.0
      IF(NCONTR.EQ.0) GO TO 200
      DO 100 IP=1,NCONTR
      X1 = XCONT1(IP)
      X2 = XCONT2(IP)
      Y1 = YCONT1(IP)
      Y2 = YCONT2(IP)
      DO 50 IJ = 1,10
      X3 = (X1 + X2)/2.0
      Y3 = (Y1 + Y2)/2.0
      D1 = (ABS(X1-X(N))**2.0+ABS(Y1-Y(N))**2.0)**0.5
      DCONT = (ABS(X3-X(N))**2.0+ABS(Y3-Y(N))**2.0)**0.5
      D2 = (ABS(X2-X(N))**2.0+ABS(Y2-Y(N))**2.0)**0.5
      IF (D1.GT.D2) GO TO 49
      X2 = X3
      Y2 = Y3
      GO TO 50
   49 X1 = X3
      Y1 = Y3
   50 CONTINUE
      IF(DCONTM.GT.DCONT) DCONTM = DCONT
  100 CONTINUE
      SCDCON = DCONTM/WXXX**(1.0/3.0)
      IF(SCDCON.GT.SCRTOP) GO TO 200
      DO 80 IB = NP2,1,-1
      IF(SCRAD(IB).LE.SCDCON) GO TO 80
      IF(IB.EQ.NP2) GO TO 75
      DEL = (DELTAR(IB+1)+DELTAR(IB))/2.0
      GO TO 81
   75 DEL = (DELTAR(IB)+1.0)/2.0
      GO TO 81
   80 CONTINUE
   81 TOP = DEL*EFACTR*WXXX**(1.0/3.0) -
     X (((WXXX/RHOE)/(DHOLE**2.0*PI/4.0))/2.0)/12.0
```

```
      IF(MSYSUN.EQ.1)
     XTOP = DEL*EFACTR*WXXX**(1.0/3.0) -
     X (((WXXX/(RHOE*0.001))/(DHOLE**2.0*PI/4.0))/2.0)/100.0
  200 CONTINUE
      RETURN
      END
      SUBROUTINE VIBRM(IV,INDEXV,K,ISOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C   PURPOSE  :                                                         C
C       TO EXAMINE VIBRATONS FOR EACH DELAY PERIOD                     C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES : REFER TO SUBROUTINE VBRCHK
C
C
      CHARACTER*1 KKODEI,SEQ
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTR
     X ,NTOTAL,MSYSUN
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),
     XSCROP,DOPTB(50)
      COMMON/DELT/DELTO,EFACTR,DELTOB,EFACTB
      COMMON/ISEQ/ IPER(50),KK,NR,NCH(50),NDECKS
      COMMON/UPDATE/XXX(50,20),YYY(50,20),ZZZ(50,20),WUPD(50,20)
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,
     X STDEPT
      DIMENSION ALOAD(50),PPV(10),PPVPIL(10),WEIGMX(50),ISOUT(50,20)
C-----------------------------------------------------------------------
      INDEXV = 0
      WEIGMX(1) = 1000000.0
      XM = 0.0
      YM = 0.0
      ZM = 0.0
      WM = 0.0
      DO 100 KC = 1,K
      DO 90 I = 1,NUMBH
      IF (ISOUT(I,KC).NE.IV) GO TO 90
      XM = XM + XXX(I,KC)*WUPD(I,KC)
      YM = YM + YYY(I,KC)*WUPD(I,KC)
      ZM = ZM + ZZZ(I,KC)*WUPD(I,KC)
      WM = WM + WUPD(I,KC)
   90 CONTINUE
  100 CONTINUE
      IF(WM.EQ.0) GO TO 200
      XM = XM / WM
      YM = YM / WM
      ZM = ZM / WM
C-----------------------------------------------------------------------
```

```
C  **  CALCULATE DISTANCE FROM POINTS  **                                LLM21460
C-----------------------------------------------------------------------LLM21470
      IF (NCHK.EQ.0) GO TO 131                                           LLM21480
      DO 130 I = 1, NCHK                                                 LLM21490
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),XM,YM,ZM,DISTM)             LLM21500
C-----------------------------------------------------------------------LLM21510
C  **   CALCULATE PPV AT THIS POINT   **                                LLM21520
C-----------------------------------------------------------------------LLM21530
      PPV(I) = AKFAC*(DISTM/WM*(1.0/2.0)) ** AMFAC                       LLM21540
C-----------------------------------------------------------------------LLM21550
C  **  FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT  **                LLM21560
C-----------------------------------------------------------------------LLM21570
      WEIGMX(I) = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DISTM**2.0              LLM21580
  130 CONTINUE                                                           LLM21590
C-----------------------------------------------------------------------LLM21600
C  **       SAME FOR PILLARS                                             LLM21610
C-----------------------------------------------------------------------LLM21620
  131 IF (NUMPIL.EQ.0) GO TO 151                                         LLM21630
      DO 150 I = 1, NUMPIL                                               LLM21640
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL2(I),YPIL2(I),ZPIL2(I),LLM21650
     X XPIL3(I),YPIL3(I),ZPIL3(I),XM,YM,ZM,DISTM)                        LLM21660
      PPVPIL(I) = AKFAC*((DISTM)/WM**(1.0/2.0))**AMFAC                   LLM21670
      WEIGMX(NCHK+I) = (AKFAC/PPVPAL)**(2.0/AMFAC)*DISTM**2.0            LLM21680
  150 CONTINUE                                                           LLM21690
C-----------------------------------------------------------------------LLM21700
C  **    FIND MINIMUM OF ACCEPTABLE WEIGHTS    **                        LLM21710
C-----------------------------------------------------------------------LLM21720
  151 WMIN = WEIGMX(1)                                                   LLM21730
      NN = NCHK + NUMPIL                                                 LLM21740
      IF (NN.EQ.0) GO TO 200                                             LLM21750
      DO 180 I = 1,NN                                                    LLM21760
      IF(WEIGMX(I).LT.WMIN) WMIN = WEIGMX(I)                             LLM21770
  180 CONTINUE                                                           LLM21780
      IF (WM.LE.(WMIN+1.0)) GO TO 200                                    LLM21790
      INDEXV = 1                                                         LLM21800
  200 RETURN                                                             LLM21810
      END                                                                LLM21820
      SUBROUTINE BENGR(X,Y,XMAX,XMIN,YMAX,YMIN,NUMBH,PLELEV)             LLM21830
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC LLM21840
C                                                                     C LLM21850
C    PURPOSE :                                                        C LLM21860
C        TO PROVIDE GRAPHICS OUTPUT                                   C LLM21870
C          LANGUAGE : DI3000                                          C LLM21880
C                                                                     C LLM21890
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC LLM21900
C                                                                      LLM21910
C                                                                      LLM21920
      CHARACTER*4 XALPHA                                                 LLM21930
      CHARACTER*2 XALP                                                   LLM21940
      CHARACTER*32 STOPID                                                LLM21950
      COMMON/IBLK/ IBLOCK(100,100),ITIME(100,100)                        LLM21960
      COMMON/XBLK/ XB(100),YB(100)                                       LLM21970
      COMMON/DXDY/ DX,DY                                                 LLM21980
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,           LLM21990
     X STDEPT                                                            LLM22000
```

```
      DIMENSION X(50),Y(50),XO(20),YO(20),XX(4),YY(4)
      REWIND 51
      CALL JBEGIN
      CALL JDINIT(1)
      CALL JDEVON(1)
      CALL JDCOLR(7)
      XM = XMIN - 10.0
      XMM = XMAX + 10.0
      YM = YMIN - 10.0
      YMM = YMAX + 10.0
      XSIZE = (XMAX-XMIN)/70.0
      YSIZE = (YMAX-YMIN)/70.0
      CALL JWINDU(XM,XMM,YM,YMM)
      CALL JOPEN
      CALL JCOLOR(6)
      CALL JPINTR(1)
      CALL JPIDEX(6,10)
      CALL JRECT(XM,YM,XMM,YMM)
      CALL JCOLOR(3)
      CALL JPINTR(1)
      CALL JPIDEX(3,10)
      CALL JRECT(XMIN,YMIN,XMAX,YMAX)
      CALL JMOVE(XMIN,YMIN)
      DO 110 J = 1,49
      KGR = 1
      DO 100 I = 1,49
      IF(IBLOCK(I,J).EQ.0) GO TO 10
      IF(IBLOCK(I,J).NE.0) GO TO 20
      GO TO 100
   10 CALL JCOLOR(3)
      CALL JPINTR(1)
      CALL JPIDEX(3,1)
      GO TO 99
   20 CALL JCOLOR(4)
      CALL JPINTR(1)
      CALL JPIDEX(4,12)
   99 IF(IBLOCK(I,J).NE.IBLOCK(I+1,J)) GO TO 200
      GO TO 150
  200 IF(KGR.EQ.1) GO TO 340
      IF(I.EQ.KGR) GO TO 433
      XX(1) = XB(KGR)-DX/2.
      YY(1) = YB(J)
      IF(J.EQ.1) YY(1)=YMIN
      XX(2) = XB(I)+DX/2.
      YY(2) = YY(1)
      IF(KGR.LE.2) GO TO 221
      IF(IBLOCK(KGR-2,J+1).EQ.IBLOCK(KGR,J)) GO TO 220
  221 IF(IBLOCK(KGR-1,J+1).EQ.IBLOCK(KGR,J)) GO TO 240
      IF(IBLOCK(KGR,J+1).EQ.IBLOCK(KGR,J)) GO TO 250
      IF(IBLOCK(KGR+1,J+1).EQ.IBLOCK(KGR,J)) GO TO 260
      IF(KGR.GE.48) GO TO 222
      IF(IBLOCK(KGR+2,J+1).EQ.IBLOCK(KGR,J)) GO TO 270
  222 XX(4) = XB(KGR+1)-DX/2.0
      YY(4) = YB(J+1)
      GO TO 300
```

```
  220 XX(4) = XB(KGR-2)-DX/2.0
      YY(4) = YB(J+1)
      GO TO 300
  240 XX(4) = XB(KGR-1)-DX/2.0
      YY(4) = YB(J+1)
      GO TO 300
  250 XX(4) = XB(KGR)-DX/2.0
      YY(4) = YB(J+1)
      GO TO 300
  260 XX(4) = XB(KGR+1) -DX/2.
      YY(4) = YB(J+1)
      GO TO 300
  270 XX(4) = XB(KGR+2) -DX/2.
      YY(4) = YB(J+1)
      GO TO 300
  340 XX(1)=XMIN
      YY(1) = YB(J)
      IF(J.EQ.1) YY(1) = YMIN
      XX(2) = XB(I)+DX/2.0
      YY(2) = YY(1)
      XX(4) = XMIN
      YY(4) = YB(J+1)
  300 IF(I.EQ.49) GO TO 350
      IF(I.GE.48) GO TO 3001
      IF(IBLOCK(I+2,J+1).EQ.IBLOCK(I,J)) GO TO 385
 3001 IF(IBLOCK(I+1,J+1).EQ.IBLOCK(I,J)) GO TO 380
      IF(IBLOCK(I,J+1).EQ.IBLOCK(I,J)) GO TO 350
      IF(IBLOCK(I-1,J+1).EQ.IBLOCK(I,J)) GO TO 360
      IF(I.LE.2) GO TO 3002
      IF(IBLOCK(I-2,J+1).EQ.IBLOCK(I,J)) GO TO 365
 3002 XX(3) = XB(I-1)+ DX/2.
      YY(3) = YB(J+1)
      GO TO 400
  385 XX(3) = XB(I+2)+DX/2.0
      YY(3) = YB(J+1)
      GO TO 400
  380 XX(3) = XB(I+1)+DX/2.0
      YY(3) = YB(J+1)
      GO TO 400
  350 XX(3) = XB(I) + DX/2.0
      YY(3) = YB(J+1)
      GO TO 400
  360 XX(3) = XB(I-1)+DX/2.0
      YY(3) = YB(J+1)
      GO TO 400
  365 XX(3) = XB(I-2)+DX/2.0
      YY(3) = YB(J+1)
  400 IF(IBLOCK(I,J).EQ.0) GO TO 411
      CALL JPOLGN(XX,YY,4)
  411 KGR = I + 1
  433 IF(I.EQ.49) GO TO 100
  150 IF(I.LT.49) GO TO 100
      GO TO 200
  100 CONTINUE
  110 CONTINUE
```

```
      CALL JCOLOR(8)
      IPREV = 0
      CALL JLWIDE(32000)
      REWIND 72
      READ (72,800) NPERP
  800 FORMAT(I5)
      READ (72,810) ((XO(I),YO(I)),I = 1,NPERP)
  810 FORMAT(2F10.2)
      DO 700 I = 1,NPERP
      KK1 = 0
      IF((XO(I).GT.XMAX).OR.(XO(I).LT.XMIN).OR.(YO(I).GT.YMAX).OR.
     X(YO(I).LT.YMIN)) GO TO 701
      IF(I.NE.1) GO TO 699
      CALL JMOVE(XO(I),YO(I))
      GO TO 700
  699 IF(IPREV.EQ.(I-1)) GO TO 710
      CALL JDRAW(XO(I),YO(I))
      GO TO 700
  701 IF(IPREV.NE.(I-1)) GO TO 710
      IPREV = I
      GO TO 700
  710 X3 = (XO(I-1)+XO(I))/2.0
      Y3 = (YO(I-1)+YO(I))/2.0
  712 KK1 = KK1 + 1
      IF(IPREV.EQ.(I-1)) GO TO 950
      IF((X3.GT.XMAX).OR.(X3.LT.XMIN).OR.(Y3.GT.YMAX).OR.(Y3.LT.YMIN))
     X GO TO 720
      X3 = (X3+XO(I))/2.0
      Y3 = (Y3+YO(I))/2.0
      IF(KK1.LT.4) GO TO 712
      GO TO 740
  720 X3 = (XO(I-1)+X3)/2.0
      Y3 = (YO(I-1)+Y3)/2.0
      IF(KK1.LT.4) GO TO 712
  740 IF(IPREV.NE.(I-1)) GO TO 760
  912 KK1 = KK1 + 1
  950 IF((X3.GT.XMAX).OR.(X3.LT.XMIN).OR.(Y3.GT.YMAX).OR.(Y3.LT.YMIN))
     X GO TO 920
      X3 = (X3+XO(I-1))/2.0
      Y3 = (Y3+YO(I-1))/2.0
      IF(KK1.LT.4) GO TO 912
      GO TO 940
  920 X3 = (XO(I)+X3)/2.0
      Y3 = (YO(I)+Y3)/2.0
      IF(KK1.LT.4) GO TO 912
  940 CALL JMOVE(X3,Y3)
      CALL JDRAW(XO(I),YO(I))
      GO TO 700
  760 IPREV = I
      CALL JDRAW(X3,Y3)
  700 CONTINUE
      CALL JCOLOR(1)
      CALL JMOVE(XMIN,YMIN)
      CALL JDRAW(XMAX,YMIN)
      CALL JMOVE(XMIN,YMIN)
```

```
      CALL JDRAW(XMIN,YMAX)                                             LLM23660
      DO 4000 I = 1,10                                                  LLM23670
      XL = XMIN + DX * FLOAT(I-1) * 10.0                                LLM23680
      IF(XL.GT.XMAX) GO TO 401                                          LLM23690
      CALL JMOVE(XL,YMIN)                                               LLM23700
      CALL JDRAW(XL,YMIN-0.5)                                           LLM23710
      CALL JMOVE(XL,YMIN-1.0)                                           LLM23720
      IF(XL.GE.0) GO TO 112                                             LLM23730
      NL = IFIX(XL - 0.5)                                               LLM23740
      GO TO 113                                                         LLM23750
  112 NL = IFIX(XL + 0.5)                                               LLM23760
  113 WRITE(61,307) NL                                                  LLM23770
      BACKSPACE 61                                                      LLM23780
      READ(61,301) XALPHA                                               LLM23790
  301 FORMAT(A4)                                                        LLM23800
      CALL JJUST(2.3)                                                   LLM23810
      CALL JSIZE(XSIZE,YSIZE)                                           LLM23820
      CALL J3STRG(XALPHA)                                               LLM23830
 4000 CONTINUE                                                          LLM23840
  401 CONTINUE                                                          LLM23850
      DO 500 I = 1,10                                                   LLM23860
      YL = YMIN + DY*FLOAT(I-1)*10.0                                    LLM23870
      IF (YL.GT.YMAX) GO TO 501                                         LLM23880
      CALL JMOVE(XMIN,YL)                                               LLM23890
      CALL JDRAW(XMIN-0.5,YL)                                           LLM23900
      CALL JMOVE(XMIN-1.0,YL)                                           LLM23910
      IF(YL.GE.0.0) GO TO 412                                           LLM23920
      NL = IFIX(YL - 0.5)                                               LLM23930
      GO TO 413                                                         LLM23940
  412 NL = IFIX(YL + 0.5)                                               LLM23950
  413 WRITE(61,307) NL                                                  LLM23960
  307 FORMAT(I4)                                                        LLM23970
      BACKSPACE 61                                                      LLM23980
      READ(61,301) XALPHA                                               LLM23990
      CALL JJUST(3.2)                                                   LLM24000
      CALL JSIZE(XSIZE,YSIZE)                                           LLM24010
      CALL J3STRG(XALPHA)                                               LLM24020
  500 CONTINUE                                                          LLM24030
  501 CALL JCMARK(3)                                                    LLM24040
      DO 600 I = 1,NUMBR                                                LLM24050
      WRITE(61,510) I                                                   LLM24060
  510 FORMAT(I2)                                                        LLM24070
      BACKSPACE 61                                                      LLM24080
      READ(61,520) XALP                                                 LLM24090
  520 FORMAT(1A2)                                                       LLM24100
      CALL JMARK(X(I),Y(I))                                             LLM24110
      CALL JMOVE(X(I)+0.6,Y(I))                                         LLM24120
      CALL JJUST(1,2)                                                   LLM24130
      CALL JSIZE(XSIZE,YSIZE)                                           LLM24140
      CALL J3STRG(XALP)                                                 LLM24150
  600 CONTINUE                                                          LLM24160
      XSIZE = 2.0*XSIZE                                                 LLM24170
      YSIZE = 2.0*YSIZE                                                 LLM24180
      REWIND 51                                                         LLM24190
      READ(51,610) STOPID                                               LLM24200
```

```
  610 FORMAT(A32)
      CALL JMOVE((XM+XMM)/2.0,YMM-3.0*(YMAX-YMIN)/70.0)
      CALL JJUST(2,2)
      CALL JSIZE(XSIZE,YSIZE)
      CALL J3STRG(STOPID)
      IELEV = IFIX(PLELEV)
      WRITE(61,630) IELEV
  630 FORMAT(I4)
      BACKSPACE 61
      READ(61,640) XALPHA
  640 FORMAT(A4)
      CALL JMOVE((XM+XMM-12.)/2.0,YMM-6.0*(YMAX-YMIN)/70.)
      CALL JJUST(2,2)
      CALL JSIZE(XSIZE,YSIZE)
      CALL J3STRG('PLAN ON ELEVATION ')
      CALL JMOVE((XM+XMM+18*XSIZE)/2.0,YMM-6.0*(YMAX-YMIN)/70.0)
      CALL JJUST(2,2)
      CALL JSIZE(XSIZE,YSIZE)
      CALL J3STRG(XALPHA)
      CALL JMOVE((XM+XMM)/2.0,YM+3.0)
      CALL JJUST(2,2)
      CALL JSIZE(XSIZE,YSIZE)
      CALL J3STRG('X AXIS (FT)')
      CALL JBASE(0.0,1.0,0.0)
      CALL JMOVE(XM+3.0,(YM+YMM)/2.0)
      CALL JJUST(2,2)
      CALL JSIZE(XSIZE,YSIZE)
      CALL J3STRG('Y AXIS (FT)')
      CALL JCLOSE
      CALL JPAUSE(1)
      CALL JDEVOF(1)
      CALL JDEND(1)
      CALL JEND
      RETURN
      END
      SUBROUTINE VBRCK2 (IFIRST,INDEXV,KKODE1,ICODE1 )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  PURPOSE :
C              TO EXAMINE THE VIBRATIONS RESULTING FROM THE DETONATIONS C
C              OF THE MODIFIED CHARGES .                                C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES  :
C         REFER TO SUBROUTINE VBRCHK
C
      CHARACTER*1 KKODEI,SEQ
      COMMON /RDRDT1/ DCONCH,WTCHRG,WSTRTH,SCDEPT(50),DELTA(50),
     XSCRAD(50),DHOLE,X(50),Y(50),Z(50),XVCHK(10),YVCHK(10),ZVCHK(10)
     X,PPVALL(10),AKFAC,AMFAC,XPIL1(10),XPIL2(10),XPIL3(10),YPIL1(10),
     XYPIL2(10),YPIL3(10),ZPIL1(10),ZPIL2(10),ZPIL3(10),PPVPAL,WSTR,
     XXMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN,DELTAR(50),RHOE,DCONCB,SCDEPB(50),
     XDELTAB(50),SCRADB(50),DELTBR(50),WTCHRB,ALENGT,BMIN,SEPDIS
      COMMON/RDRDT2/ NP1,NPB1,NP2,NPB2,NUMBH,NCHK,NUMPIL,INIT(50),NCONTR
```

```
     X ,NTOTAL,MSYSUN                                                   LLM24760
      COMMON/CHARGE/WEIGHT(50),WEIGB(50),DOPT(50),RCR(50),DEPTH(50),    LLM24770
     XSCROP,DOPTB(50)                                                   LLM24780
      COMMON/DELT/DELTU,EFACTR,DELTOB,EFACTB                            LLM24790
      COMMON/ISEW/ IPER(50),KK,NR,NCH(50),NDECKS                        LLM24800
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,          LLM24810
     X STDEPT                                                           LLM24820
      DIMENSION ALOAD(50),PPV(10),PPVPIL(10),WEIGMX(50),ID(50),TOP(50)  LLM24830
C-----------------------------------------------------------------------LLM24840
C  **  INITIALIZE VARIABLES  **                                         LLM24850
C-----------------------------------------------------------------------LLM24860
      DO 2 I = 1,50                                                     LLM24870
      ID(I) = 0                                                         LLM24880
    2 CONTINUE                                                          LLM24890
      WEIGMX(1) = 100000.0                                              LLM24900
      INDEXV = 0                                                        LLM24910
C-----------------------------------------------------------------------LLM24920
C  **  FIND CHARGES DETONATING IN THE SAME PERIOD  **                   LLM24930
C-----------------------------------------------------------------------LLM24940
      KK = 0                                                            LLM24950
      DO 100  IA = 1, NUMBH                                             LLM24960
      IF (IPER(IA).NE.IFIRST) GO TO 100                                 LLM24970
C-----------------------------------------------------------------------LLM24980
C  **          LOAD HOLE           **                                  LLM24990
C-----------------------------------------------------------------------LLM25000
      PI = 3.14159                                                      LLM25010
      IF (NTOTAL.NE.NDECKS) GO TO 20                                    LLM25020
      CALL TOPH(IA,TOP(IA))                                             LLM25030
      ALOAD(IA) = (12.0*(SEPDIS+ALLNGT-TOP(IA))*PI*DHOLE**2.0/4.0)*RHOE LLM25040
      IF(MSYSUN.EQ.1) ALOAD(IA) = ALOAD(IA)/12.0*0.1                    LLM25050
C-----------------------------------------------------------------------LLM25060
C  **   CHECK VIBRATION LEVELS   **                                    LLM25070
C-----------------------------------------------------------------------LLM25080
C                                                                       LLM25090
   20 IF(NCHK.EQ.0) GO TO 32                                            LLM25100
C-----------------------------------------------------------------------LLM25110
C  **  CALCULATE DISTANCES FROM POINTS                                  LLM25120
C-----------------------------------------------------------------------LLM25130
      DO 30 I = 1, NCHK                                                 LLM25140
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),X(IA),Y(IA),Z(IA),DIST)    LLM25150
C-----------------------------------------------------------------------LLM25160
C  **  CALCULATE PPV AT THIS POINT  **                                  LLM25170
C-----------------------------------------------------------------------LLM25180
      PPV(I) = AKFAC*(DIST/WEIGHT(IA)**(1.0/2.0))**AMFAC                LLM25190
C-----------------------------------------------------------------------LLM25200
C  ** FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT **                  LLM25210
C-----------------------------------------------------------------------LLM25220
      WEIGMX(I) = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DIST**2.0              LLM25230
   30 CONTINUE                                                          LLM25240
C-----------------------------------------------------------------------LLM25250
C  **  CALCULATE DISTANCE FROM PILLARS  **                              LLM25260
C-----------------------------------------------------------------------LLM25270
   32 IF (NUMPIL.EQ.0) GO TO 52                                         LLM25280
      DO 50 I = 1, NUMPIL                                               LLM25290
      CALL DISTPL(XPIL1(I),YPIL1(I),ZPIL1(I),XPIL2(I),YPIL2(I),ZPIL2(I),LLM25300
```

```
     XXPIL3(I),YPIL3(I),ZPIL3(I),X(IA),Y(IA),Z(IA),DISTP)
C-----------------------------------------------------------------------
C  ** CALCULATE PPV AT THIS POINT **
C-----------------------------------------------------------------------
      PPVPIL(I) = AKFAC * (DISTP/WEIGHT(IA)**(1.0/2.0))**
     X AMFAC
C-----------------------------------------------------------------------
C  ** FIND MAXIMUM WEIGHT BECAUSE OF THIS PARTICULAR PILLAR **
C-----------------------------------------------------------------------
      WEIGMX(NCHK+I) = (AKFAC/PPVPAL)**(2.0/AMFAC)*DISTP**2.0
   50 CONTINUE
C-----------------------------------------------------------------------
C  ** FIND MINIMUM ACCEPTABLE WEIGHTS **
C-----------------------------------------------------------------------
   52 WMIN = WEIGMX(1)
      NN = NCHK + NUMPIL
      IF (NN.EQ.0) GO TO  99
      DO 80 I = 1, NN
      IF (WEIGMX(I).LT.WMIN) WMIN=WEIGMX(I)
   80 CONTINUE
      IF (WEIGHT(IA).LE.WMIN) GO TO 99
      WEIGHT(IA) = WMIN
   99 KK = KK + 1
      INIT(KK) = IA
  100 CONTINUE
      IF(NN.EQ.0) GO TO 200
C-----------------------------------------------------------------------
C       CHECK VIBRATIONS DUE TO THE SIMULTANEOUS  DETONATION OF
C                 MANY CHARGES
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C  ** FIND AN AVERAGE POSITION OF THE CONCENTRATED CHARGE **
C-----------------------------------------------------------------------
      XM = 0.0
      IF(KK.EQ.1) GO TO 200
      YM = 0.0
      ZM = 0.0
      WM = 0.0
      DO 110  I = 1,KK
      IM = INIT(I)
      XM = XM + X(I)*WEIGHT(IM)
      YM = YM + Y(I)*WEIGHT(IM)
      ZM = ZM + Z(I)*WEIGHT(IM)
      WM = WM + WEIGHT(IM)
  110 CONTINUE
      XM = XM / WM
      YM = YM / WM
      ZM = ZM / WM
C-----------------------------------------------------------------------
C  ** CALCULATE DISTANCE FROM POINTS **
C-----------------------------------------------------------------------
      IF (NCHK.EQ.0) GO TO 131
      DO 130 I = 1, NCHK
      CALL DISTPT(XVCHK(I),YVCHK(I),ZVCHK(I),XM,YM,ZM,DISTM)
```

```
C-----------------------------------------------------------------------------
C  **   CALCULATE PPV AT THIS POINT   **
C-----------------------------------------------------------------------------
      PPV(I) = AKFACA(DISTM/WMA(1.0/2.0)) ** AMFAC
C-----------------------------------------------------------------------------
C  **  FIND MAXIMUM WEIGHT FOR THIS PARTICULAR POINT  **
C-----------------------------------------------------------------------------
      WEIGMX(I) = (AKFAC/PPVALL(I))**(2.0/AMFAC)*DISTM**2.0
  130 CONTINUE
C-----------------------------------------------------------------------------
C  **       SAME FOR PILLARS
C-----------------------------------------------------------------------------
  131 IF (NUMPIL.EQ.0) GO TO 151
      DO 150 I = 1. NUMPIL
      CALL DISTPL(XPIL1(I).YPIL1(I),ZPIL1(I),XPIL2(I).YPIL2(I),ZPIL2(I).
     X XPIL3(I).YPIL3(I),ZPIL3(I),XM,YM,ZM,DISTM)
      PPVPIL(I) = AKFACA((DISTM)/WMA*(1.0/2.0))**AMFAC
      WEIGMX(NCHK+I) = (AKFAC/PPVPAL)**(2.0/AMFAC)*DISTM**2.0
  150 CONTINUE
C-----------------------------------------------------------------------------
C  **    FIND MINIMUM OF ACCEPTABLE WEIGHTS    **
C-----------------------------------------------------------------------------
  151 WMIN = 2000000.0
      NN = NCHK + NUMPIL
      IF (NN.EQ.0) GO TO 200
      DO 180 I = 1,NN
      IF(WEIGMX(I).LT.WMIN) WMIN = WEIGMX(I)
  180 CONTINUE
      IF (WM.LE.WMIN) GO TO 200
      IF (KKODEI.NE.'N') GO TO 192
      IF ( ICODE1.EQ.1 )GO TO 199
      WRITE(7,187)
  187 FORMAT('  THE NUMBER OF CHARGES DETONATING IN THIS PERIOD IS'./
     X'  TOO LARGE RESULTING IN UNACCEPTABLE VIBRATIONS.',/
     X'  DO YOU WANT TO CHANGE THE SEQUENCING OF THE CHARGES ? (Y/N)')
      READ (7,188) SEQ
  188 FORMAT(1A1)
      INDEXV = 1
      IF(SEQ.EQ.'Y') GO TO 191
      INDEXV = 0
      GO TO 200
  192 INDEXV = 1
  191 DO 190 I = 1,NUMBH
      IF(IPER(I).NE.IFIRST) GO TO 190
      WEIGHT(I) = 0.0
      WEIGB(I) = 0.0
      IPER(I) = 100000
  190 CONTINUE
      GO TO 200
  199 INDEXV = 1
      DO 2190 I = 1,NUMBH
      IF(IPER(I).NE.IFIRST) GO TO 2190
      WEIGHT(I) = 0.0
      WEIGB(I) = 0.0
 2190 CONTINUE
```

```
  200 RETURN
      END
      SUBROUTINE CHARGL(DDUUPD)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     PURPOSE :                                                        C
C         TO PREPARE THE DATA FILE FOR THE GRAPHICS PROGRAM            C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES : REFER TO MAIN PROGRAM
C
C
      CHARACTER*80 TITLE
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,
     X STDEPT
      DIMENSION ZKNOWN(50,20),ALENGT(50,20),WEIGHT(50,20),IPER(50,20),
     X XDI(50),YDI(50),ZDI(50),STOPID(40),DDUUPD(50,20)
      PP = 3.1415926
      REWIND 70
      READ(70,910) NUMBH,NCHRGS
  910 FORMAT(15I5)
      READ(70,900) DIAMT
      DO 1000 I = 1,NCHRGS
      READ(70,900) (ZKNOWN(J,I),J=1,NUMBH)
  900 FORMAT(8F10.2)
 1000 CONTINUE
      DO 1050 I = 1,NCHRGS
      READ(70,900) (WEIGHT(J,I),J=1,NUMBH)
 1050 CONTINUE
      DO 1100 I = 1,NCHRGS
      READ(70,900) (ALENGT(J,I),J=1,NUMBH)
 1100 CONTINUE
      IEXP = 1
      DO 1200 I = 1,NCHRGS
      READ(70,910) (IPER(J,I),J=1,NUMBH)
 1200 CONTINUE
      REWIND 66
      DO 3000 IHOSEQ = 1,NUMBH
      READ(66,100) IJUNK,NI,IHOROW,IHOCOL,DDDDD,DIPNOM
  100 FORMAT(4I5,2F10.2)
      IF(IJUNK.EQ.99) GO TO 3008
      WRITE(62,21) NCHRGS,IHOROW,IHOCOL
   21 FORMAT(15I5)
      READ(66,101) XCOLAR,YCOLAR,ZCOLAR
  101 FORMAT(3F10.2)
      READ(66,101,END=3008) ((XDI(I),YDI(I),ZDI(I)),I=1,NI)
      DO 155 ICH = 1,NCHRGS
      X1 = XCOLAR
      Y1 = YCOLAR
      Z1 = ZCOLAR-ZREFER
      D1 = 0
      DO 150 I = 1,NI
      Z2 = ZDI(I)-ZREFER
      X2 = XDI(I)
```

```
      Y2 = YDI(I)
      IF(Z2.LT.ZKNOWN(IHOSEQ,ICH)) GO TO 109
      ZLOCAL = ZKNOWN(IHOSEQ,ICH)
      RATIO = (ZKNOWN(IHOSEQ,ICH)-Z1)/(Z2-Z1)
      DX1X2 = RATIO*(X1-X2)
      DY1Y2 = RATIO*(Y1-Y2)
      DI=DI+(ABS(DX1X2)**2.0+ABS(DY1Y2)**2.0+ABS(Z1-ZKNOWN(IHOSEQ,ICH))
     X**2.0)**0.5
      GO TO 200
  109 ZLOCAL = Z2
      DI=DI + (ABS(X1-X2)**2.0+ABS(Y1-Y2)**2.0+ABS(Z1-Z2)**2.0)**0.5
      X1 = X2
      Y1 = Y2
      Z1 = Z2
  150 CONTINUE
  200 DUP = DI - ALENGT(IHOSEQ,ICH)/2.0
      DDO = DI + ALENGT(IHOSEQ,ICH)/2.0
      DDOUPD(IHOSEQ,ICH) = DDO
      ZDO = ZCOLAR-ZREFER-(ZKNOWN(IHOSEQ,ICH)+ALENGT(IHOSEQ,ICH)/2.0)
      ZUP = ZCOLAR-ZREFER-(ZKNOWN(IHOSEQ,ICH)-ALENGT(IHOSEQ,ICH)/2.0)
      WRITE(62,260) ZDO,ZUP,DDO,DUP,WEIGHT(IHOSEQ,ICH),DIAMT,IEXPLT,IPER
     X(IHOSEQ,ICH)
  260 FORMAT(6F10.2,2I5)
  155 CONTINUE
 3000 CONTINUE
 3008 CONTINUE
      LAST = 99
      WRITE(62,21) LAST,IHOROW,IHOCOL
      RETURN
      END
      SUBROUTINE PLVIEW(PLELEV,XHOL,YHOL,NOHOLS)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C   PURPOSE  :                                                        C
C               TO GIVE THE  LOCAL COORDINATES OF THE HOLES AT        C
C               A SPECIFIED ELEVATION                                 C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C        PLELEV : PLAN ELEVATION
C        XREFER , YREFER , ZREFER . REFERENCE POINT FOR TRANSLATION
C
C
C
C
C
C
      COMMON/REFER/ XREFER,YREFER,ZREFER,STANGR,STLENG,STWIDT,
     X STDEPT
      DIMENSION D(50),DIP(50),AZIM(50)
      DIMENSION X(50),Y(50),Z(50)
      DIMENSION STOPID(50)
      DIMENSION XHOL(50),YHOL(50)
      DIMENSION HOL(50,3),XP(50),YP(50),DIAM(50)
```

```
      PP=3.1415926                                                      LLM27510
      REWIND 66                                                         LLM27520
      PLELEV=PLELEV-ZREFER                                              LLM27530
      IF (PLELEV.LT.0.0) RETURN                                         LLM27540
    6 FORMAT(I1,',',I1)                                                 LLM27550
   19 FORMAT(20A4)                                                      LLM27560
   20 FORMAT(4F10.2)                                                    LLM27570
C-----------------------------------------------------------------------LLM27580
C   READ HOLE DATA                                                      LLM27590
C    99 DENOTES END OF HOLE DATA FILE                                   LLM27600
C-----------------------------------------------------------------------LLM27610
C                                                                       LLM27620
    7 READ (66,100) IHOSEQ,N1,IHOROW,IHOCOL,DIA,DIPNUM                  LLM27630
      IF (IHOSEQ.EQ.99) GO TO 8                                         LLM27640
C-----------------------------------------------------------------------LLM27650
C   INCREASE HOLE DIAMETER BY *200 FOR PLOT CLARITY                     LLM27660
C                                                                       LLM27670
      READ (66,101) XCOLAR,YCOLAR,ZCOLAR                                LLM27680
C-----------------------------------------------------------------------LLM27690
C       STATION INTERVAL  10 FT                                         LLM27700
C      FOLLOWING DATA REFERS TO HOLE AXIS                               LLM27710
C-----------------------------------------------------------------------LLM27720
      READ (66,101,END=8) ((X(I),Y(I),Z(I)),I=1,N1)                     LLM27730
      CALL PTRANS(X,Y,Z,N1,XREFER,YREFER,ZREFER)                        LLM27740
C-----------------------------------------------------------------------LLM27750
C    ROTATE COORDINATES (STANGR)                                        LLM27760
C-----------------------------------------------------------------------LLM27770
      DO 200 I = 1,N1                                                   LLM27780
      XC =   X(I) * COS(STANGR) + Y(I) * SIN (STANGR)                   LLM27790
      YC = - X(I) * SIN(STANGR) + Y(I) * COS (STANGR)                   LLM27800
      X(I) = XC                                                         LLM27810
      Y(I) = YC                                                         LLM27820
      HOL(I,1) = X(I)                                                   LLM27830
      HOL(I,2) = Y(I)                                                   LLM27840
      HOL(I,3) = Z(I)                                                   LLM27850
  200 CONTINUE                                                          LLM27860
C-----------------------------------------------------------------------LLM27870
C  **   ROTATE COORDINATES   **                                         LLM27880
C-----------------------------------------------------------------------LLM27890
      DO 10 I=2,N1                                                      LLM27900
      IF (HOL(I,3).LT.PLELEV) GO TO 10                                  LLM27910
      XP(1)=HOL(I,1)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,  LLM27920
     *1)-HOL(I,1))                                                      LLM27930
      YP(1)=HOL(I,2)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,  LLM27940
     *2)-HOL(I,2))                                                      LLM27950
      GO TO 11                                                          LLM27960
   10 CONTINUE                                                          LLM27970
   11 CONTINUE                                                          LLM27980
      XP(2)=HOL(1,1)                                                    LLM27990
      YP(2)=HOL(1,2)                                                    LLM28000
      XHOL(IHOSEQ) = XP(1)                                              LLM28010
      YHOL(IHOSEQ) = YP(1)                                              LLM28020
      GO TO 7                                                           LLM28030
    8 CONTINUE                                                          LLM28040
  100 FORMAT(4I5,2F10.2)                                                LLM28050
```

```
 101  FORMAT(3F10.2)
      REWIND 51
      REWIND 66
      RETURN
      END
      SUBROUTINE PTRANS(X,Y,Z,NPNTS,XREF,YREF,ZREF)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     PURPOSE  :                                                       C
C        TO PERFORM A TRANSLATION WRT THE POINT XREF,YREF,ZREF         C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION X(50),Y(50),Z(50)
      DO 10 I=1,NPNTS
      X(I)=X(I)-XREF
      Y(I)=Y(I)-YREF
      Z(I)=Z(I)-ZREF
   10 CONTINUE
      RETURN
      END
```

3. GRAPHICS OUTPUT GENERATOR PROGRAM FOR BOTH DESIGN PROGRAMS

```
      PROGRAM PGRAPH
C-----------------------------------------------------------------------
C     MENU DRIVER  FOR PLOTTING..
C        AXONOMETRIC PROJECTION OF A STOPE, PLANS AND CROSS SECTIONS WITH
C        THE HOLE LOCATIONS AND SINGLE HOLE PROJECTIONS
C-----------------------------------------------------------------------
      WRITE (6,100)
  100 FORMAT (2X,'LARGE DIAMETER BLASTHOLE MINING'/
     *10X,'G R A P H I C S'/)
      WRITE(6,1133)
 1133 FORMAT(2X,'ENTER  1 FOR GRAPHICS OUTPUT FOR A VCR STOPE',/
     X2X,'    OR  2 FOR GRAPHICS OUTPUT FOR AN UNDERGROUND BENCH.')
      READ(6,*) IGRAPH
  111 WRITE (6,112)
  112 FORMAT (2X,'DO YOU WISH TO DRAW'/
     *3X,'(1) A PLAN ON A CERTAIN ELEVATION'/
     *3X,'(2) A CROSS SECTION'/
     *3X,'(3) A SINGLE HOLE'/
     *3X,'(4) EXIT')
      READ(5,*,ERR=111,END=111) MM
      IF (MM.LT.1.OR.MM.GT.4) GO TO 111
      GO TO (20,30,40,50),MM
C-----------------------------------------------------------------------
C     DRAW AN AXONOMETRIC PROJECTION
C-----------------------------------------------------------------------
C
   20 WRITE (6,21)
   21 FORMAT (2X,'ENTER THE ELEVATION')
      READ(5,*,END=20) PLELEV
C-----------------------------------------------------------------------
C     DRAW AN PLAN ON ELEVATION PLELEV
C-----------------------------------------------------------------------
      CALL PLAN(PLELEV)
      GO TO 111
C
C-----------------------------------------------------------------------
C     DRAW A COLUMN OR ROW SECTION ON COLUMN OR ROW # IROW
C-----------------------------------------------------------------------
   30 WRITE (6,31)
   31 FORMAT (2X,'ENTER (1) FOR COLUMN SECTION'/
     *2X,' OR   (2) FOR ROW SECTION')
      READ(5,*,END=30) IRC
      IF (IRC.GT.2.OR.IRC.LT.1) GO TO 30
   32 IF (IRC.EQ.2) WRITE (6,33)
      IF (IRC.EQ.1) WRITE (6,34)
   33 FORMAT (2X,'ENTER  ROW NUMBER')
   34 FORMAT (2X,'ENTER  COLUMN NUMBER')
      READ(5,*,END=32) IROW
      CALL PCROSN(IROW,IRC,IGRAPH)
      GO TO 111
C-----------------------------------------------------------------------
C     DRAW A SINGLE HOLE PROJECTION OF HOLE (IROW,ICOLUM)
C-----------------------------------------------------------------------
   40 WRITE (6,41)
   41 FORMAT (2X,'ENTER THE ROW #, COLUMN #')
```

```
      READ(5,*,END=40) IROW,ICOLUM                                      PGR00560
      CALL PHOPRU(IROW,ICOLUM)                                          PGR00570
      GO TO 111                                                         PGR00580
   50 CONTINUE                                                          PGR00590
      STOP                                                              PGR00600
      END                                                               PGR00610
C                                                                       PGR00620
C     CALCULATES THE ADJUSTED HOLE BOUNDARIES. FOR- RA                  PGR00630
C     IT ALSO CALCULATES THE AZIMUTH OF THE PERPENDICULAR TO A SEGMENT  PGR00640
C                                                                       PGR00650
C      SLOP : VECTOR DIRECTION (LOCAL)                                  PGR00660
C      SLOP : VECTOR DIRECTION +PP/2.  (RETURN)                         PGR00670
C      X,Y  : HOLE AXIS                                                 PGR00680
C      RA,  : HOLE RADIUS                                               PGR00690
C      XA,YA,XB,YB  : HOLE WALLS PLUS OR MINUS RADIUS                   PGR00700
C                                                                       PGR00710
      SUBROUTINE PANG(X,Y,N,RA)                                         PGR00720
      DIMENSION ANGL(50),X(50),Y(50)                                    PGR00730
      COMMON/BL1/XA(50),YA(50),XB(50),YB(50),SLOP(50)                   PGR00740
      PP=3.1415927                                                      PGR00750
      N1=N-1                                                            PGR00760
      N2=N-2                                                            PGR00770
      DO 10 I=1,N1                                                      PGR00780
      FI=0.0                                                            PGR00790
      IF ((X(I+1)-X(I)).EQ.0.0) FI=PP/2.                                PGR00800
      IF ((X(I+1)-X(I)).NE.0.0) FI=ATAN((Y(I)-Y(I+1))/(X(I)-X(I+1)))    PGR00810
      IF (FI.EQ.0.0.AND.X(I).LT.X(I+1)) FI=FI+PP                        PGR00820
      IF (FI.LT.0.0) FI=FI+PP                                           PGR00830
      IF ((Y(I)-Y(I+1)).LT.0.0) FI=FI+PP                                PGR00840
      SLOP(I)=FI                                                        PGR00850
   10 CONTINUE                                                          PGR00860
      DO 20 I=2,N1                                                      PGR00870
      AS=SLOP(I-1)+PP                                                   PGR00880
      IF (AS.GT.2.0*PP) AS=AS-2.*PP                                     PGR00890
      ANGL(I)=SLOP(I)-AS                                                PGR00900
      IF (ANGL(I).GT.2.0*PP) ANGL(I)=ANGL(I)-2.*PP                      PGR00910
      IF (ANGL(I).LT.0.0) ANGL(I)=ANGL(I)+2.*PP                         PGR00920
   20 CONTINUE                                                          PGR00930
C                                                                       PGR00940
C     CALCULATE WALLS                                                   PGR00950
C                                                                       PGR00960
      DO 30 I=2,N1                                                      PGR00970
      AL=RA                                                             PGR00980
      FI2=ANGL(I)/2.                                                    PGR00990
      IF (SIN(FI2).NE.0.0) AL=RA/SIN(FI2)                               PGR01000
      FI3=SLOP(I-1)+FI2                                                 PGR01010
      XA(I)=X(I)+AL*COS(FI3)                                            PGR01020
      YA(I)=Y(I)+AL*SIN(FI3)                                            PGR01030
      XB(I)=X(I)-AL*COS(FI3)                                            PGR01040
      YB(I)=Y(I)-AL*SIN(FI3)                                            PGR01050
   30 CONTINUE                                                          PGR01060
C                                                                       PGR01070
C ADD POINTS N AND 1                                                    PGR01080
C                                                                       PGR01090
      AL=RA                                                             PGR01100
```

```
      FI3=SLOP(N-1)+PP/2.
      XA(N)=X(N)+ALACOS(FI3)
      YA(N)=Y(N)+ALASIN(FI3)
      XB(N)=X(N)-ALACOS(FI3)
      YB(N)=Y(N)-ALASIN(FI3)
      FI3=2.0*PP-SLOP(1)
      XA(1)=X(1)+AL/SIN(FI3)
      YA(1)=Y(1)
      XB(1)=X(1)-AL/SIN(FI3)
      YB(1)=Y(1)
C
C     MAKE DIRECTION +PP/2. (USED ONLY FOR HORIZONTAL PROJECTION DEPTH)
C
      DO 40 I=1,N1
      SLOP(I)=SLOP(I)+PP/2.
   40 CONTINUE
      RETURN
      END
      SUBROUTINE PROJE2 (XYZ,N,NAXIS,X,Y,Z,XMAX,YMAX,ZMAX)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C      DEPENDING ON THE X,Y,Z SEQUENCE PROJECT THE 3-D COORDINATES    C
C      OF XYZ TO X,Y                                                  C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DIMENSION XYZ(50,3),X(50),Y(50),Z(50)
      XMAX=0.0
      YMAX=0.0
      ZMAX=0.0
      IF (N.EQ.0) RETURN
      DO 20 I=1,N
      X(I)=XYZ(I,1)
      Y(I)=XYZ(I,2)
      Z(I)=XYZ(I,3)
      CHE=ABS(X(I))
      IF (XMAX.LT.CHE) XMAX=CHE
      CHE=ABS(Y(I))
      IF (YMAX.LT.CHE) YMAX=CHE
      CHE=ABS(Z(I))
      IF (ZMAX.LT.CHE) ZMAX=CHE
   20 CONTINUE
  102 FORMAT (3F8.2)
      RETURN
      END
      SUBROUTINE PTRANS(X,Y,Z,NPNTS,XREF,YREF,ZREF)
C----------------------------------------------------------------------
C  PERFORM A TRANSLATION WRT THE POINT XREF,YREF,ZREF
C----------------------------------------------------------------------
      DIMENSION X(50),Y(50),Z(50)
      DO 10 I=1,NPNTS
      X(I)=X(I)-XREF
      Y(I)=Y(I)-YREF
      Z(I)=Z(I)-ZREF
   10 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE PHOPRO (IRO,ICO)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C   PROJECTS SINGLE HOLES AT THE THREE MAIN PLANES                     C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CHARACTER*4 AX1
      CHARACTER*32 STOPID
      DIMENSION HOL(50,3),X(50),Y(50),Z(50),NN1(4),XP(50),YP(50)
      COMMON/BL1/XA(50),YA(50),XB(50),YB(50),DIRE(50)
      PP=3.1415927
C
      REWIND 51
      REWIND 61
      READ (51,19) STOPID
      READ (51,20) XREFER,YREFER,ZREFER
      READ (51,20) STLENG,STWIDT,STDEPT
      READ (51,20) STANGR
      STANGR = STANGR*PP/180.
      READ (51,100) NOHOLS
    6 FORMAT(I1,',',I1)
   19 FORMAT(A32)
   20 FORMAT(4F10.2)
C
    7 READ (61,100) IHOSEQ,N1,IHORO,IHOCO,DIA,DIPNUM
      IF (IHOSE.EQ.99) WRITE (6,133)
  133 FORMAT ('  HOLE NOT FOUND')
      IF (IHOSE.EQ.99) GO TO 999
C-----------------------------------------------------------------------
C   INCREASE HOLE DIAMETER BY X200 FOR PLOT CLARITY
C
      DIA=DIA/400.0
      RA=DIA/2.0
      READ (61,101) XCOLAR,YCOLAR,ZCOLAR
C-----------------------------------------------------------------------
C        STATION INTERVAL  10 FT
C       FOLLOWING DATA REFERS TO HOLE AXIS
C-----------------------------------------------------------------------
      READ (61,101,END=999) ((X(I),Y(I),Z(I)),I=1,N1)
      IF (IHORO.NE.IRO.OR.IHOCO.NE.ICO) GO TO 7
C
      CALL PTRANS(X,Y,Z,N1,XCOLAR,YCOLAR,ZCOLAR)
C-----------------------------------------------------------------------
C    ROTATE COORDINATES (STANGR)
C-----------------------------------------------------------------------
      DO 200 I = 1,N1
      XC =   X(I) * COS(STANGR) + Y(I) * SIN (STANGR)
      YC = - X(I) * SIN(STANGR) + Y(I) * COS (STANGR)
      X(I) = XC
      Y(I) = YC
      HOL(I,1) = X(I)
      HOL(I,2) = Y(I)
      HOL(I,3) = - Z(I)
  200 CONTINUE
```

```
C
  100 FORMAT(4I5,2F10.2)
  101 FORMAT(3F10.2)
C
  108 WRITE (6,109)
  109 FORMAT (2X,'GIVE # OF PROJECTIONS AND RESPECTIVE AXES'/
     +2X,'(3) HORIZONTAL'/
     +2X,'(2) PLANE XZ  '/
     +2X,'(1) PLANE YZ ')
       READ (5,X,END=108) NPRO,(NN1(I),I=1,NPRO)
C
       DO 130 IH1=1,NPRO
       NAXIS=NN1(IH1)
C
C      GET X,Y ARRAYS FOR PLOT AND THE XTENT OF 'DEVIATION'
C
       IF (NAXIS.EQ.3) CALL PROJE2(HOL,N1,NAXIS,X,Y,Z,XMAX,YMAX,ZMAX)
       IF (NAXIS.EQ.2) CALL PROJE2(HOL,N1,NAXIS,X,Z,Y,XMAX,ZMAX,YMAX)
       IF (NAXIS.EQ.1) CALL PROJE2(HOL,N1,NAXIS,Z,X,Y,ZMAX,XMAX,YMAX)
       XWP=XMAX
       YWP=YMAX
       IF (XMAX.LT.4.0) XWP=4.0
       IF (YMAX.LT.4.0) YWP=4.0
       XW=XWP*2.0
       XW2=XWP*2.0
       YW2=YWP*2.0
       YW=YWP*1.30
       IF (NAXIS.NE.3) YW2=.25*YWP
C-----------------------------------------------------------------
C      CALL CMSCMD('EXEC LINEMODE ON.',IRC)
C
       CALL JBEGIN
       CALL JDINIT ( 1)
       CALL JDEVON ( 1)
       CALL JDPINT (1)
       CALL JDJUST (2,2)
       CALL JDINTE (32767)
         CALL JDPIDX(2,0)
         CALL JDLSTY(0)
         CALL JDCOLR(7)
         CALL JSETER(7)
         CALL JVSPAC(-1.,1.,-1.,1.)
        CALL  JVPORT (-1.,1.,-1.,1.)
        CALL JWINDO(-XW,XW,-YW,YW2)
         CALL JWCLIP (.FALSE.)
C-----------------------------------------------------------
C LETTER SIZE CALCULATIONS
C-----------------------------------------------------------
       XLETR=4.0*XWP*0.04*0.7
       YLETR=(YW+YW2)*0.04
       XLETR2=XLETR/2.
       YLETR2=YLETR/2.
C
       CALL JOPEN
       CALL JPIDEX (7.0)
```

```
      CALL JMOVE (-XW.-YW)                                              PGR02760
      CALL JRRECT (2.0*XW,YW-YW2)                                       PGR02770
C                                                                       PGR02780
C   DRAW SCALE MARKS ALONG THE TWO AXES                                 PGR02790
C                                                                       PGR02800
      CALL JMARK(3)                                                     PGR02810
C   AXES.DRAW  X OR HORIZONTAL                                          PGR02820
       X1=-1.25*XWP                                                     PGR02830
       Y1=0.                                                            PGR02840
       CALL JMOVL(X1.Y1)                                                PGR02850
       X1=-X1                                                           PGR02860
      CALL JCOLOR(2)                                                    PGR02870
       CALL JDRAW(X1,Y1)                                                PGR02880
       N13=13                                                           PGR02890
       DO 15 L=1,N13                                                    PGR02900
       XP(L)=X1-(L-1)*X1/6.0                                            PGR02910
   15 YP(L)=0.0                                                         PGR02920
      CALL JCOLOR(1)                                                    PGR02930
       CALL JPMARK(XP,YP,N13)                                           PGR02940
       X1=-X1                                                           PGR02950
       Y1=-YLETR2                                                       PGR02960
       CALL JMOVE(X1,Y1)                                                PGR02970
       CALL JSIZE(XLLTR2,YLLTR2)                                        PGR02980
       CALL JJUST(1,3)                                                  PGR02990
       IF(NAXIS.EQ.2.OR.NAXIS.EQ.3) CALL J3STRG('-X')                   PGR03000
       IF(NAXIS.EQ.1) CALL J3STRG('-Y')                                 PGR03010
  901 FORMAT(F4.1)                                                      PGR03020
  904 FORMAT(F4.0)                                                      PGR03030
  902 FORMAT(A4)                                                        PGR03040
  903 FORMAT(I4)                                                        PGR03050
       N11=N13-2                                                        PGR03060
       CALL JJUST(2,3)                                                  PGR03070
       DO 16 L=3,N11,2                                                  PGR03080
       IF (L.EQ.7) GO TO 16                                             PGR03090
       X1=XP(L)                                                         PGR03100
       Y1=-YLETR2                                                       PGR03110
       X2=ABS(X1)                                                       PGR03120
       WRITE(92,901) X2                                                 PGR03130
       BACKSPACE 92                                                     PGR03140
       READ(92,902) AX1                                                 PGR03150
       CALL JMOVE(X1,Y1)                                                PGR03160
       CALL J3STRG(AX1)                                                 PGR03170
   16 CONTINUE                                                          PGR03180
C                                                                       PGR03190
       X1=1.25*XWP                                                      PGR03200
       CALL JMOVE(X1,Y1)                                                PGR03210
       CALL JJUST(3,3)                                                  PGR03220
       IF(NAXIS.EQ.2.OR.NAXIS.EQ.3) CALL J3STRG('X')                    PGR03230
       IF(NAXIS.EQ.1) CALL J3STRG('Y')                                  PGR03240
       IF(NAXIS.EQ.3) GO TO 136                                         PGR03250
C                                                                       PGR03260
C VERTICAL AXIS  (Z FOR OPTONS 1 + 2)                                   PGR03270
C                                                                       PGR03280
      X1=0.                                                             PGR03290
       Y1=0.                                                            PGR03300
```

```
      CALL JMOVE(Y1,X1)                                                 PGR03310
      X1=-1.15*YWP                                                      PGR03320
     CALL JCOLOR(2)                                                     PGR03330
      CALL JDRAW(Y1,X1)                                                 PGR03340
      N4=4                                                              PGR03350
      DO 225 L=1,N4                                                     PGR03360
      XP(L)=-YMAX+(L-1)*YMAX/4.0                                        PGR03370
  225 YP(L)=0.0                                                         PGR03380
     CALL JCOLOR(1)                                                     PGR03390
      CALL JPMARK(YP,XP,N4)                                             PGR03400
      X1=X1                                                             PGR03410
      Y1=-XLETR2                                                        PGR03420
      CALL JMOVE(Y1,X1)                                                 PGR03430
      CALL JJUST(3,2)                                                   PGR03440
      IF(NAXIS.EQ.2.OR.NAXIS.EQ.1) CALL J3STRG('-Z')                    PGR03450
C     IF(NAXIS.EQ.3) CALL J3STRG('-Y')                                  PGR03460
C                                                                       PGR03470
      CALL JJUST(3,2)                                                   PGR03480
      DO 226 L=1,N4                                                     PGR03490
      X1=XP(L)                                                          PGR03500
      X2=ABS(X1)                                                        PGR03510
      WRITE(92,904) X2                                                  PGR03520
      BACKSPACE 92                                                      PGR03530
      READ(92,902) AX1                                                  PGR03540
      CALL JMOVE(-XLETR2,X1)                                            PGR03550
      CALL J3STRG(AX1)                                                  PGR03560
  226 CONTINUE                                                          PGR03570
      GO TO 137                                                         PGR03580
C                                                                       PGR03590
C   Y AXIS  FOR OPTION 3                                                PGR03600
C                                                                       PGR03610
  136 X1=-1.25*YWP                                                      PGR03620
      Y1=0.                                                             PGR03630
      CALL JMOVE(Y1,X1)                                                 PGR03640
      X1=-X1                                                            PGR03650
     CALL JCOLOR(2)                                                     PGR03660
      CALL JDRAW(Y1,X1)                                                 PGR03670
      N13=13                                                            PGR03680
      N11=N13-2                                                         PGR03690
      DO 25 L=1,N13                                                     PGR03700
      XP(L)=X1-(L-1)*X1/6.0                                             PGR03710
   25 YP(L)=0.0                                                         PGR03720
     CALL JCOLOR(1)                                                     PGR03730
      CALL JPMARK(YP,XP,N13)                                            PGR03740
      X1=-X1                                                            PGR03750
      Y1=-XLETR2                                                        PGR03760
      CALL JMOVE(Y1,X1)                                                 PGR03770
      CALL JJUST(3,2)                                                   PGR03780
      IF(NAXIS.EQ.2.OR.NAXIS.EQ.1) CALL J3STRG('-Z')                    PGR03790
      IF(NAXIS.EQ.3) CALL J3STRG('-Y')                                  PGR03800
C                                                                       PGR03810
      CALL JJUST(3,2)                                                   PGR03820
      DO 26 L=3,N11,2                                                   PGR03830
      IF (L.EQ.7) GO TO 26                                              PGR03840
      X1=XP(L)                                                          PGR03850
```

```
      Y1=-XLETR2                                                        PGR03860
      X2=ABS(X1)                                                        PGR03870
      WRITE(92,901) X2                                                  PGR03880
      BACKSPACE 92                                                      PGR03890
      READ(92,902) AX1                                                  PGR03900
      CALL JMOVE(Y1,X1)                                                 PGR03910
      CALL JSSTRG(AX1)                                                  PGR03920
   26 CONTINUE                                                          PGR03930
C                                                                       PGR03940
      X1=1.25*YWP                                                       PGR03950
      CALL JMOVE(Y1,X1)                                                 PGR03960
      CALL JJUST(3,2)                                                   PGR03970
      IF(NAXIS.EQ.2.OR.NAXIS.EQ.1) CALL JSSTRG('Z')                     PGR03980
      IF(NAXIS.EQ.3) CALL JSSTRG('Y')                                   PGR03990
C                                                                       PGR04000
  137 CONTINUE                                                          PGR04010
C                                                                       PGR04020
C  CALL PANG TO FIND THE DIRECTIONS OF THE DEPTH INDICATORS             PGR04030
C                                                                       PGR04040
      CALL PANG (X,Y,N1,KA)                                             PGR04050
C                                                                       PGR04060
C  DRAW HOLE  AXIS ONLY     (IF TWO LINES ARE TO BE PLOTTED IT GETS     PGR04070
C                            TOO MESSY)                                 PGR04080
C                                                                       PGR04090
      CALL JMOVE(X(1),Y(1))                                             PGR04100
       CALL JCOLOR(1)                                                   PGR04110
        CALL JPOLY(X,Y,N1)                                              PGR04120
C                                                                       PGR04130
      IF (NAXIS.NE.3) GO TO 132                                         PGR04140
      DZ=HOL(N1,3)-HOL(1,3)                                             PGR04150
      DZ1=50.                                                           PGR04160
      IF (ABS(DZ).LT.120.0) DZ1=20.                                     PGR04170
      IF (ABS(DZ).LT.60.) DZ1=10.                                       PGR04180
      DZ1=SIGN(DZ1,DZ)                                                  PGR04190
      NZ=DZ/DZ1                                                         PGR04200
C                                                                       PGR04210
C     INDICATE DEPTH ON  HORIZONTAL  PROJECTION ONLY                    PGR04220
C                                                                       PGR04230
      DO 110 J=1,NZ                                                     PGR04240
      DZ=DZ1*J                                                          PGR04250
      DO 111 I=1,N1                                                     PGR04260
     IF (HOL(I,3).GT.DZ) GO TO 111                                      PGR04270
     XA(J)=X(I)+(DZ-HOL(1,3))/(HOL(I-1,3)-HOL(1,3))*(X(I-1)-X(I))       PGR04280
     XB(J)=XA(J)+XLETR2*COS(DIRE(I))                                    PGR04290
     YA(J)=Y(I)+(DZ-HOL(1,3))/(HOL(I-1,3)-HOL(1,3))*(Y(I-1)-Y(I))       PGR04300
     YB(J)=YA(J)+YLETR2*SIN(DIRE(I))                                    PGR04310
     CALL JMOVL(XA(J),YA(J))                                            PGR04320
     CALL JDRAW(XB(J),YB(J))                                            PGR04330
C                                                                       PGR04340
      AN=DIRE(I)                                                        PGR04350
      IF (AN.LT. 45..OR. AN.GE.315.) CALL JJUST (1,2)                   PGR04360
      IF (AN.GT. 45..AND.AN.LT.135.) CALL JJUST (2,1)                   PGR04370
      IF (AN.GE.135..AND.AN.LT.225.) CALL JJUST (3,2)                   PGR04380
      IF (AN.GE.225..AND.AN.LT.315.) CALL JJUST (2,3)                   PGR04390
      ID2=INT(DZ)                                                       PGR04400
```

```
       WRITE(92,903) IB2
       BACKSPACE 92
       READ(92,902) AX1
       CALL J3STRG(AX1)
      GO TO 112
  111 CONTINUE
  112 CONTINUE
  110 CONTINUE
  132 CONTINUE
C-------------------------------------------------------------------------
C      TITTLE
C-------------------------------------------------------------------------
       X1=0.0
       Y1=0.15*YWP
       IF (NAXIS.EQ.3) Y1=1.85*YWP
       CALL JMOVE(X1,Y1)
       CALL JJUST(2,2)
       CALL JSIZE(XLETR,YLETR)
       CALL JGAP(0.0)
       CALL JCOLOR(4)
       IF (NAXIS.EQ.3) CALL J3STRG('HORIZONTAL PROJECTION')
       IF (NAXIS.EQ.2) CALL J3STRG('PROJECTION ON PLANE XZ')
       IF (NAXIS.EQ.1) CALL J3STRG('PROJECTION ON PLANE YZ')
      CALL JCLOSE
      CALL JPAUSE(1)
      CALL JDEVOF(1)
      CALL JDEND(1)
      CALL JEND
  130 CONTINUE
  999 CONTINUE
      RETURN
      END
      SUBROUTINE PLAN (PLELEV)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C  THIS ROUTINE PLOTS A PLAN VIEW AT A SPECIFIED ELEVATION              C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C        PLELEV : PLAN ELEVATION
C        STOPID : STOPE IDENTIFICATION (UP TO 80 CHARACTERS)
C       XREFER , YREFER , ZREFER . REFERENCE POINT FOR TRANSLATION
C       XLETER,YLETER : LETER SIZE
C       XLETR2,YLETR2 : 1/2 LETER SIZE
C
C
C
C
C
C
C
      CHARACTER*3 AA2
      CHARACTER*4 AA1,CEL1
      CHARACTER*32 STOPID
      DIMENSION D(50),DIP(50),AZIM(50)
```

```
      DIMENSION X(50),Y(50),Z(50)
      DIMENSION HOL(50,3),XP(50),YP(50),DIAM(50)
      PP=3.1415926
      REWIND 51
      REWIND 61
      READ (51,19) STOPID
      READ (51,20) XREFER,YREFER,ZREFER
      READ (51,20) STLENG,STWIDT,STDEPT
      READ (51,20) STANGR
      STANGR = STANGR*PP/180.
      READ (51,100) NOHOLS
      PLELEV=PLELEV-ZREFER
      IF (PLELEV.LT.0.0) RETURN
    6 FORMAT(I1,',',I1)
   19 FORMAT(A32)
   20 FORMAT(4F10.2)
C-----------------------------------------------------------------
C WINDOW DIMENSION CALCULATIONS
C-----------------------------------------------------------------
      XWIND = STLENG
      IF (STWIDT.GT.STLENG) XWIND = STWIDT
      YWIND = XWIND
C-----------------------------------------------------------------
C     CALL CMSCMD('EXEC LINEMODE ON.',IRC)
C
      CALL JBEGIN
      CALL JDINIT ( 1)
      CALL JDEVON ( 1)
      CALL JDPINT (1)
      CALL JDJUST (2,2)
      CALL JDINTE (32767)
        CALL JDPIDX(7,0)
        CALL JDLSTY(0)
        CALL JDCOLR(7)
        CALL JSETER(7)
        CALL JVSPAC(-1.,1.,-1.,1.)
       CALL  JVPORT (-0.8,0.8,-.8,0.8)
        CALL JWINDO(0.0,XWIND,0.0,YWIND)
        CALL JWCLIP (.FALSE.)
C-----------------------------------------------------------------
C LETTER SIZE CALCULATIONS
C-----------------------------------------------------------------
      XLETR=.04*XWIND
      YLETR=.04*XWIND*.7
      XLETR2=XLETR/2.
      YLETR2=YLETR/2.
C
      CALL JOPEN
      CALL JPIDEX (7,0)
      CALL JMOVE (-0.1*XWIND,-0.1*YWIND)
      CALL JRRECT (1.2*XWIND,1.2*YWIND)
      CALL JPIDEX (2,0)
      CALL JMOVE (0.0,0.0)
      CALL JRRECT (STLENG,STWIDT)
C
```

```
C    DRAW SCALE MARKS ALONG THE TWO AXES                                PGR05510
C                                                                       PGR05520
      IM=INT(STWIDT/10.)+1                                              PGR05530
      DO 400 I=1,IM                                                     PGR05540
      X(I)=0.                                                           PGR05550
      Y(I)=(I-1)A10.                                                    PGR05560
  400 CONTINUE                                                          PGR05570
      CALL JJUST(3,2)                                                   PGR05580
      CALL JCOLOR(1)                                                    PGR05590
      CALL JSIZE(XLETR2,YLETR2)                                         PGR05600
      DO 401 I=1,IM                                                     PGR05610
      X55=X(I)-XLETR2                                                   PGR05620
      Y55=Y(I)                                                          PGR05630
      CALL JMOVE(X55,Y55)                                               PGR05640
      AW=(I-1)A10.                                                      PGR05650
      WRITE (92,102) AW                                                 PGR05660
      BACKSPACE 92                                                      PGR05670
      READ (92,103) AA1                                                 PGR05680
      CALL J3STRG(AA1)                                                  PGR05690
      X55=X(I)-XLETR2/2.                                                PGR05700
      Y55=Y(I)                                                          PGR05710
      CALL JMOVE(X55,Y55)                                               PGR05720
      CALL JDRAW(X(I),Y(I))                                             PGR05730
  401 CONTINUE                                                          PGR05740
      IM=INT(STLENG/10.)+1                                              PGR05750
      DO 405 I=1,IM                                                     PGR05760
      Y(I)= 0.                                                          PGR05770
      X(I)=(I-1)A10.                                                    PGR05780
  405 CONTINUE                                                          PGR05790
      CALL JJUST(2,3)                                                   PGR05800
      DO 406 I=1,IM                                                     PGR05810
      X55=X(I)                                                          PGR05820
      Y55=Y(I)-YLETR2                                                   PGR05830
      CALL JMOVE(X55,Y55)                                               PGR05840
      AW=(I-1)A10.                                                      PGR05850
      WRITE (92,102) AW                                                 PGR05860
      BACKSPACE 92                                                      PGR05870
      READ (92,103) AA1                                                 PGR05880
      CALL J3STRG(AA1)                                                  PGR05890
      X55=X(I)                                                          PGR05900
      Y55=Y(I)-YLETR2/2.                                                PGR05910
      CALL JMOVE(X55,Y55)                                               PGR05920
      CALL JDRAW(X(I),Y(I))                                             PGR05930
  406 CONTINUE                                                          PGR05940
C-----------------------------------------------------------------------PGR05950
C  MARKS DRAWN                                                          PGR05960
C-----------------------------------------------------------------------PGR05970
C    READ HOLE DATA                                                     PGR05980
C     99 DENOTES END OF HOLE DATA FILE                                  PGR05990
C-----------------------------------------------------------------------PGR06000
C                                                                       PGR06010
    7 READ (61,100) IHOSEQ,N1,IHOROW,IHOCOL,DIA,DIPNOM                  PGR06020
      IF (IHOSEQ.EQ.99) GO TO 8                                         PGR06030
C-----------------------------------------------------------------------PGR06040
C    INCREASE HOLE DIAMETER BY X200 FOR PLOT CLARITY                    PGR06050
```

```
C                                                                       PGR06060
      DIA=DIA/200.0                                                     PGR06070
      READ (61,101) XCOLAR,YCOLAR,ZCOLAR                                PGR06080
C-----------------------------------------------------------------------PGR06090
C        STATION INTERVAL  10 FT                                        PGR06100
C       FOLLOWING DATA REFERS TO HOLE AXIS                              PGR06110
C-----------------------------------------------------------------------PGR06120
      READ (61,101,END=8) ((X(I),Y(I),Z(I)),I=1,N1)                     PGR06130
      CALL PTRANS(X,Y,Z,N1,XREFER,YREFER,ZREFER)                        PGR06140
C-----------------------------------------------------------------------PGR06150
C    ROTATE COORDINATES (STANGR)                                        PGR06160
C-----------------------------------------------------------------------PGR06170
      DO 200 I = 1,N1                                                   PGR06180
      XC =   X(I) * COS(STANGR) + Y(I) * SIN (STANGR)                   PGR06190
      YC = - X(I) * SIN(STANGR) + Y(I) * COS (STANGR)                   PGR06200
      X(I) = XC                                                         PGR06210
      Y(I) = YC                                                         PGR06220
      HOL(I,1) = X(I)                                                   PGR06230
      HOL(I,2) = Y(I)                                                   PGR06240
      HOL(I,3) = Z(I)                                                   PGR06250
  200 CONTINUE                                                          PGR06260
C-----------------------------------------------------------------------PGR06270
C   ROTATE COORDINATES                                                  PGR06280
C-----------------------------------------------------------------------PGR06290
      DO 10 I=2,N1                                                      PGR06300
      IF (HOL(I,3).LT.PLELEV) GO TO 10                                  PGR06310
      XP(I)=HOL(I,1)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,  PGR06320
     *1)-HOL(I,1))                                                      PGR06330
      YP(I)=HOL(I,2)+(PLELEV-HOL(I,3))/(HOL(I-1,3)-HOL(I,3))*(HOL(I-1,  PGR06340
     *2)-HOL(I,2))                                                      PGR06350
      GO TO 11                                                          PGR06360
   10 CONTINUE                                                          PGR06370
   11 CONTINUE                                                          PGR06380
      XP(2)=HOL(1,1)                                                    PGR06390
      YP(2)=HOL(1,2)                                                    PGR06400
      Z1=1.                                                             PGR06410
      N12=6                                                             PGR06420
      X1=XP(1)                                                          PGR06430
      X2=XP(2)                                                          PGR06440
      Y1=YP(1)                                                          PGR06450
      Y2=YP(2)                                                          PGR06460
      RA=DIA                                                            PGR06470
C-----------------------------------------------------------------------PGR06480
C TAG  HOLES ON PLOT                                                    PGR06490
C-----------------------------------------------------------------------PGR06500
      CALL JCOLOR(8)                                                    PGR06510
      X12=X1                                                            PGR06520
      Y12=Y1                                                            PGR06530
      FI=0.0                                                            PGR06540
      IF ((X1-X2).EQ.0.0) FI=PP/2.                                      PGR06550
      IF ((X1-X2).NE.0.0) FI=ATAN((Y1-Y2)/(X1-X2))                      PGR06560
      IF (FI.EQ.0.0.AND.Y2.LT.Y1) FI=FI+PP                              PGR06570
      IF (FI.LT.0.0) FI=FI+PP                                           PGR06580
      IF ((X2-X1).LT.0.0) FI=FI+PP                                      PGR06590
      WRITE (92,6) IHOKOW,IHOCOL                                        PGR06600
```

```
      BACKSPACE 92                                                      PGR0661
      READ (92,137) AA2                                                 PGR0662
      CALL JSIZE(XLETR2,YLETR2)                                         PGR0663
      CALL JGAP(0.)                                                     PGR0664
      CALL JJUST(1,2)                                                   PGR0665
      IF (X2.GT.X1) CALL JJUST(3,2)                                     PGR0666
      IF (X2.GT.X1) CALL JMOVE(X12-XLETR2/2.0,Y12)                      PGR0667
      IF (X2.LE.X1) CALL JMOVE(X12+XLETR2/2.0,Y12)                      PGR0668
      CALL J3STR6(AA2)                                                  PGR0669
C-----------------------------------------------------------------------PGR0670
C HOLE ID TAG END                                                       PGR0671
C-----------------------------------------------------------------------PGR0672
      CALL JLSTYL(0)                                                    PGR0673
      CALL JPIDEX(1.32767)                                              PGR0674
      CALL JMOVE(X1,Y1)                                                 PGR0675
      CALL JCIRCL(X1,Y1,Z1,RA,N12)                                      PGR0676
C     PERPENTICULAR                                                     PGR0677
C                                                                       PGR0678
      X3=X1+RA*COS(FI+PP/2.)                                            PGR0679
      Y3=Y1+RA*SIN(FI+PP/2.)                                            PGR0680
      CALL JMOVE(X3,Y3)                                                 PGR0681
      X4=X2+RA*COS(FI+PP/2.)                                            PGR0682
      Y4=Y2+RA*SIN(FI+PP/2.)                                            PGR0683
      CALL JLSTYL(1)                                                    PGR0684
      CALL JDRAW(X4,Y4)                                                 PGR0685
      CALL JMOVE(X2,Y2)                                                 PGR0686
      CALL JCOLOR(8)                                                    PGR0687
      CALL JPINTR(1)                                                    PGR0688
      CALL JPIDEX(8.0)                                                  PGR0689
      CALL JCIRCL(X2,Y2,Z1,RA,N12)                                      PGR0690
C-----------------------------------------------------------------------PGR0691
C DRAW AXES  ON THE PROJECTION OF TOP   TO DISTINGUISH FROM BOTTOM      PGR0692
C-----------------------------------------------------------------------PGR0693
      X3=X2-RA*1.5                                                      PGR0694
      CALL JMOVE(X3,Y2)                                                 PGR0695
      X3=X2+RA*1.5                                                      PGR0696
      CALL JLSTYL(1)                                                    PGR0697
      CALL JDRAW(X3,Y2)                                                 PGR0698
      Y3=Y2-RA*1.5                                                      PGR0699
      CALL JMOVE(X2,Y3)                                                 PGR0700
      Y3=Y2+RA*1.5                                                      PGR0701
      CALL JDRAW(X2,Y3)                                                 PGR0702
      X3=X2+RA*COS(FI+3.0*PP/2.)                                        PGR0703
      Y3=Y2+RA*SIN(FI+3.0*PP/2.)                                        PGR0704
      CALL JMOVE(X3,Y3)                                                 PGR0705
      X3=X1+RA*COS(FI+3.0*PP/2.)                                        PGR0706
      Y3=Y1+RA*SIN(FI+3.0*PP/2.)                                        PGR0707
      CALL JLSTYL(2)                                                    PGR0708
      CALL JDRAW(X3,Y3)                                                 PGR0709
C   26 CONTINUE                                                         PGR0710
      GO TO 7                                                           PGR0711
    8 CONTINUE                                                          PGR0712
C-----------------------------------------------------------------------PGR0713
C DRAW TITLE                                                            PGR0714
C-----------------------------------------------------------------------PGR0715
```

```
      CALL JCOLOR(1)
      CALL JJUST(1,1)
      CALL JSIZE(XLETR,YLETR)
      CALL JGAP(0.)
      Y1=1.1*YWINDA*.95
      X1=0.0
      CALL JMOVE(X1,Y1)
      CALL J3STRG(STOPID)
      PL=PLELEV+ZREFER
      WRITE (92,102) PL
      BACKSPACE 92
      READ (92,103) CEL1
      Y3=Y1-1.5*YLETR
      CALL JMOVE(X1,Y3)
      CALL J3STRG('PLAN ON ELEVATION')
      X3=17.5*XLETR
      CALL JMOVE(X3,Y3)
      CALL J3STRG(CEL1)
  100 FORMAT(4I5,2F10.2)
  101 FORMAT(3F10.2)
  102 FORMAT(F6.1)
  103 FORMAT(2A4)
  107 FORMAT(A4)
  137 FORMAT(A3)
      CALL JCLOSE
      CALL JPAUSE (1)
      CALL JDEVOF (1)
      CALL JDEND (1)
      CALL JEND
      RETURN
      END
      SUBROUTINE PCROSN (IHRC,IRC)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C  PURPOSE :                                                           C
C    TO GENERATE A CROS SECTION ON A ROW OR                            C
C    COLUMN BASIS                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   VARIABLES :
C     IRC=1 COLUMN BASIS (REFERENCE)
C     IRC=2 ROW    BASIS (REFERENCE)
C     IROW= THE ROW OR COLUMN NUMBER
C
C        STOPID : STOPE IDENTIFICATION (UP TO 80 CHARACTERS)
C        XREFER , YREFER , ZREFER . REFERENCE POINT FOR TRANSLATION
C        XLETER,YLETER : LETER SIZE
C        XLETR2,YLETR2 : 1/2 LETER SIZE
C
C
      CHARACTER*32 STOPID
      CHARACTER*4 AA1
      CHARACTER*2 AA2
      DIMENSION X(50),Y(50),Z(50)
```

```
      DIMENSION HOL(50,3)                                               PGR07710
      DIMENSION XX(40),YY(40)                                           PGR07720
      DIMENSION MAT(21)                                                 PGR07730
      COMMON/BL1/XA(50),YA(50),XB(50),YB(50),DIRE(50)                   PGR07740
      PP=3.1415926                                                      PGR07750
   61 WRITE (6,65)                                                      PGR07760
   65 FORMAT(2X,'DO YOU WISH TO DRAW THE CHARGES? 1=Y')                 PGR07770
      READ (5,*,END=61) ICH                                             PGR07780
C                                                                       PGR07790
      REWIND 51                                                         PGR07800
      REWIND 61                                                         PGR07810
      READ (51,19) STOPID                                               PGR07820
      READ (51,20) XREFER,YREFER,ZREFER                                 PGR07830
      READ (51,20) STLENG,STWIDT,STDEPT                                 PGR07840
      READ (51,20) STANGR                                               PGR07850
      STANGR = STANGR*PP/180.                                           PGR07860
    6 FORMAT(I1,',',I1)                                                 PGR07870
   19 FORMAT(A32)                                                       PGR07880
   20 FORMAT(4F10.2)                                                    PGR07890
C                                                                       PGR07900
C-----------------------------------------------------------------------PGR07910
C WINDOW DIMENSION CALCULATIONS                                         PGR07920
C-----------------------------------------------------------------------PGR07930
      YWIND = STDEPT                                                    PGR07940
      XWIND = STWIDT                                                    PGR07950
      IF (IRC.EQ.1) XWIND=STLENG                                        PGR07960
C-----------------------------------------------------------------------PGR07970
C     CALL CMSCMD('EXEC LINEMODE ON.',IRCODE)                           PGR07980
C                                                                       PGR07990
      CALL JBEGIN                                                       PGR08000
      CALL JDINIT ( 1)                                                  PGR08010
      CALL JDEVON ( 1)                                                  PGR08020
      CALL JDPINT (1)                                                   PGR08030
      CALL JDJUST (2,2)                                                 PGR08040
      CALL JDINTE (32767)                                               PGR08050
        CALL JDPIDX(2.0)                                                PGR08060
        CALL JDLSTY(0)                                                  PGR08070
        CALL JDCOLK(7)                                                  PGR08080
        CALL JSETER(7)                                                  PGR08090
        CALL JVSPAC(-1.,1.,-1.,1.)                                      PGR08100
       CALL  JVPORT (-0.8,0.8,-.8,0.8)                                  PGR08110
        CALL JWINDO(0.0,XWIND,-YWIND,0.0)                               PGR08120
        CALL JWCLIP (.FALSE.)                                           PGR08130
C-----------------------------------------------------------------------PGR08140
C LETTER SIZE CALCULATIONS                                              PGR08150
C-----------------------------------------------------------------------PGR08160
      XLETR=.04*XWIND                                                   PGR08170
      YLETR=.04*YWIND*.7                                                PGR08180
      XLETR2=XLETR/2.                                                   PGR08190
      YLETR2=YLETR/2.                                                   PGR08200
      XLETR3 = XLETR2/1.3                                               PGR08210
      YLETR3 = YLETR2/1.3                                               PGR08220
C                                                                       PGR08230
      CALL JOPEN                                                        PGR08240
      CALL JPIDEX (7,0)                                                 PGR08250
```

```
      CALL JMOVE (-0.1*XWIND,-1.1*YWIND)                                PGR08260
      CALL JRRECT (1.2*XWIND,1.2*YWIND)                                 PGR08270
      CALL JPIDEX (2.0)                                                 PGR08280
      CALL JMOVE (0.0,-YWIND)                                           PGR08290
      CALL JRRECT (XWIND,YWIND)                                         PGR08300
C                                                                       PGR08310
C   DRAW SCALE MARKS ALONG THE TWO AXES                                 PGR08320
C                                                                       PGR08330
      IM=INT(YWIND/10.0)+1                                              PGR08340
      DO 400 I=1,IM                                                     PGR08350
      X(I)=0.                                                           PGR08360
      Y(I)=(1-I)*10.                                                    PGR08370
  400 CONTINUE                                                          PGR08380
      CALL JCOLOR(1)                                                    PGR08390
      CALL JJUST(3,2)                                                   PGR08400
      CALL JSIZE(XLETR2,YLETR2)                                         PGR08410
      DO 401 I=1,IM                                                     PGR08420
      CALL JMOVE(X(I)-XLETR2,Y(I))                                      PGR08430
      AW=(I-1)*10.                                                      PGR08440
      WRITE (92,110) AW                                                 PGR08450
      BACKSPACE 92                                                      PGR08460
      READ (92,103) AA1                                                 PGR08470
      CALL J3STRG(AA1)                                                  PGR08480
      CALL JMOVE(X(I)-XLETR2/2.,Y(I))                                   PGR08490
      CALL JDRAW(X(I),Y(I))                                             PGR08500
  401 CONTINUE                                                          PGR08510
C                                                                       PGR08520
      IM=INT(XWIND/10.)+1                                               PGR08530
      DO 405 I=1,IM                                                     PGR08540
      Y(I)=-YWIND                                                       PGR08550
      X(I)=(I-1)*10.                                                    PGR08560
  405 CONTINUE                                                          PGR08570
C                                                                       PGR08580
      CALL JJUST(2,3)                                                   PGR08590
      CALL JSIZE(XLETR2,YLETR2)                                         PGR08600
      DO 406 I=1,IM                                                     PGR08610
      CALL JMOVE(X(I),Y(I)-YLETR2)                                      PGR08620
      AW=(I-1)*10.                                                      PGR08630
      WRITE (92,110) AW                                                 PGR08640
      BACKSPACE 92                                                      PGR08650
      READ (92,103) AA1                                                 PGR08660
      CALL J3STRG(AA1)                                                  PGR08670
      CALL JMOVE(X(I),Y(I)-YLETR2/2.)                                   PGR08680
      CALL JDRAW(X(I),Y(I))                                             PGR08690
  406 CONTINUE                                                          PGR08700
C  MARKS DRAWN                                                          PGR08710
C                                                                       PGR08720
C-----------------------------------------------------------------------PGR08730
   40 READ (61,100) IHUSEQ,N1,IHURD,IHUCU,DIA,DIPNUM                    PGR08740
      IF (IHUSEQ.EQ.99) GO TO 41                                        PGR08750
      DIA=DIA/200.0                                                     PGR08760
      RA =DIA/2.0                                                       PGR08770
      READ (61,101) XCULAR,YCULAR,ZCULAR                                PGR08780
C-----------------------------------------------------------------------PGR08790
C       STATION INTERVAL  10 FT                                         PGR08800
```

```
C       FOLLOWING DATA REFERS TO HOLE AXIS                                PGR0881
C-----------------------------------------------------------------------PGR0882
      READ (61,101,END=41) ((X(I),Y(I),Z(I)),I=1,N1)                      PGR0883
      IF (IRC.EQ.2.AND.IHORD.NE.IHRC) GO TO 40                            PGR0884
      IF (IRC.EQ.1.AND.IHOCD.NE.IHRC) GO TO 40                            PGR0885
C-----------------------------------------------------------------------PGR0886
C  PERFORM A TRANSLATION WRT THE POINT XREF,YREF,ZREF                     PGR0887
C-----------------------------------------------------------------------PGR0888
      CALL PTRANS(X,Y,Z,N1,XREFER,YREFER,ZREFER)                          PGR0889
C-----------------------------------------------------------------------PGR0890
C    ROTATE COORDINATES (STANGR)                                          PGR0891
C-----------------------------------------------------------------------PGR0892
      DO 200 I = 1,N1                                                     PGR0893
      XC =    X(I) * COS(STANGR) + Y(I) * SIN (STANGR)                    PGR0894
      YC = - X(I) * SIN(STANGR) + Y(I) * COS (STANGR)                     PGR0895
      X(I) = XC                                                           PGR0896
      Y(I) = YC                                                           PGR0897
  200 CONTINUE                                                            PGR0898
      DO 51 I=1,N1                                                        PGR0899
      IF (IRC.EQ.2) X(I)=Y(I)                                             PGR0900
   51 Y(I)= - Z(I)                                                        PGR0901
C                                                                         PGR0902
C                                                                         PGR0903
      CALL PANG(X,Y,N1,KA)                                                PGR0904
C  DRAW HOLE                                                              PGR0905
       CALL JMOVE(XB(1),YB(1))                                            PGR0906
        CALL JPOLY(XB,YB,N1)                                              PGR0907
       CALL JMOVE(XA(1),YA(1))                                            PGR0908
        CALL JPOLY(XA,YA,N1)                                              PGR0909
C                                                                         PGR0910
C      DRAW AXIS   ONLY                                                   PGR0911
C                                                                         PGR0912
       CALL JMOVE(XA(N1),YA(N1))                                          PGR0913
       CALL JDRAW(XB(N1),YB(N1))                                          PGR0914
C ANGLES INCORPORATED                                                     PGR0915
C TAG                                                                     PGR0916
      CALL JSIZE(XLETR,YLETR)                                             PGR0917
      CALL JJUST(2,1)                                                     PGR0918
      WRITE (92,6) IHORD,IHOCD                                            PGR0919
      BACKSPACE 92                                                        PGR0920
      READ (92,103) AA1                                                   PGR0921
      CALL JMOVE(X(1),Y(1)+YLETR/4.)                                      PGR0922
      CALL J3STRG(AA1)                                                    PGR0923
C-----------------------------------------------------------------------PGR0924
C TAG END                                                                 PGR0925
C-----------------------------------------------------------------------PGR0926
C         DRAW CHARGES                                                    PGR0927
C-----------------------------------------------------------------------PGR0928
      IF (ICH.NE.1) GO TO 40                                              PGR0929
C                                                                         PGR0930
C READ CHARGE INFO      NCHRGS= CHARGES+NO OF STEMMINGS                   PGR0931
      REWIND 62                                                           PGR0932
  185 READ (62,21) NCHRGS,IRD,ICD                                         PGR0933
      IF (NCHRGS.EQ.99) GO TO 40                                          PGR0934
      DO 300 L=1,NCHRGS                                                   PGR0935
```

```
      READ (62,17) DEST,DEFI,AST,AFI,QUANT,CHDIA,IEXPLT                 PGR09360
     *,IDELT                                                            PGR09370
   21 FORMAT(15I5)                                                      PGR09380
   17 FORMAT(6F10.2,2I5)                                                PGR09390
      IF (IRO.NE.IHORO.OR.ICO.NE.IHOCO) GO TO 300                       PGR09400
      IF (QUANT.LT.0.01) GO TO 300                                      PGR09410
C    LOCATE                                                             PGR09420
      CHRA=CHDIA/400.                                                   PGR09430
C                                                                       PGR09440
      DO 70 I=1,N1                                                      PGR09450
      IF (YB(I).GT.DEST) GO TO 70                                       PGR09460
      YST=XB(I)+(DEST-YB(I))/(YB(I-1)-YB(I))*(XB(I-1)-XB(I))            PGR09470
      NYS=I-1                                                           PGR09480
      GO TO 71                                                          PGR09490
   70 CONTINUE                                                          PGR09500
   71 CONTINUE                                                          PGR09510
      DO 80 I=1,N1                                                      PGR09520
      IF (YB(I).GE.DEFI) GO TO 80                                       PGR09530
      YFI=XB(I)+(DEFI-Y(I))/(Y(I-1)-Y(I))*(XB(I-1)-XB(I))               PGR09540
      NYF=I                                                             PGR09550
      GO TO 81                                                          PGR09560
   80 CONTINUE                                                          PGR09570
   81 I1=1                                                              PGR09580
      XX(I1)=YFI                                                        PGR09590
      YY(I1)=DEFI                                                       PGR09600
      IF (NYF.GT.NYS) GO TO 90                                          PGR09610
      DO 85 I=NYF,NYS                                                   PGR09620
      I1=I1+1                                                           PGR09630
      XX(I1)=XB(I)                                                      PGR09640
      YY(I1)=YB(I)                                                      PGR09650
   85 CONTINUE                                                          PGR09660
   90 CONTINUE                                                          PGR09670
      I1=I1+1                                                           PGR09680
      XX(I1)=YST                                                        PGR09690
      YY(I1)=DEST                                                       PGR09700
C                                                                       PGR09710
      DO 170 I=1,N1                                                     PGR09720
      IF (YA(I).GT.DEST) GO TO 170                                      PGR09730
      YST=XA(I)+(DEST-YA(I))/(YA(I-1)-YA(I))*(XA(I-1)-XA(I))            PGR09740
      NYS=I-1                                                           PGR09750
      GO TO 171                                                         PGR09760
  170 CONTINUE                                                          PGR09770
  171 CONTINUE                                                          PGR09780
      DO 180 I=1,N1                                                     PGR09790
      IF (YA(I).GE.DEFI) GO TO 180                                      PGR09800
      YFI=XA(I)+(DEFI-YA(I))/(YA(I-1)-YA(I))*(XA(I-1)-XA(I))            PGR09810
      NYF=I                                                             PGR09820
      GO TO 181                                                         PGR09830
  180 CONTINUE                                                          PGR09840
  181 I1=I1+1                                                           PGR09850
      IL1=I1                                                            PGR09860
      XX(I1)=YST                                                        PGR09870
      YY(I1)=DEST                                                       PGR09880
      IF (NYF.GT.NYS) GO TO 190                                         PGR09890
      DO 1185 I=NYF,NYS                                                 PGR09900
```

```
      JJ1=NYS+NYF-I
      I1=I1+1
      XX(I1)=XA(JJ1)
      YY(I1)=YA(JJ1)
 1185 CONTINUE
  190 CONTINUE
      I1=I1+1
      XX(I1)=YFI
      YY(I1)=DEFI
      IL2=I1
      IF (XX(1).LT.XX(I1)) GO TO 197
      IL1=IL1-1
      IL2=1
C
  197 X1=XX(1)
      Y1=YY(1)
      CALL JMOVE(X1,Y1)
      MATER=2
      IF (MATER.EQ.2) MCOL = 1
      IF (MATER.EQ.1) MCOL = 6
      CALL JPIDEX(MCOL,3*MATER)
      CALL JPOLGN(XX,YY,I1)
      CALL JSIZE(XLETR3,YLETR3)
      X1=XX(IL1)
      Y1=YY(IL1)
      X2=XX(IL2)
      Y2=YY(IL2)
      CALL JMOVE(X1+XLETR3/5.,Y1)
      WRITE (92,108) AST
      BACKSPACE 92
      READ (92,103) AA1
      CALL JJUST(1,3)
      CALL J3STRG(AA1)
      WRITE (92,108) AFI
      BACKSPACE 92
      READ (92,103) AA1
      CALL JJUST(1,1)
      CALL JMOVE(X2+XLETR3/5.,Y2)
      CALL J3STRG(AA1)
      Y3=(Y2+Y1)/2.
      X3=AMAX1(X1,X2)
      CALL JMOVE(X3+XLETR3/5.,Y3+YLETR3/5.)
      WRITE (92,108) QUANT
      BACKSPACE 92
      READ (92,103) AA1
      CALL JJUST(1,1)
      CALL J3STRG(AA1)
      CALL JMOVE(X3+XLETR3/5.,Y3-YLETR3/5.0)
      WRITE (92,109) IDELT
      BACKSPACE 92
      READ (92,103) AA1
      CALL JJUST(1,3)
      CALL J3STRG(AA1)
  300 CONTINUE
C
```

```
      GO TO 185                                                         PGR10460
   41 CONTINUE                                                          PGR10470
C-----------------------------------------------------------------------PGR10480
C        TITLE                                                          PGR10490
      CALL JSIZE(XLETR,YLETR)                                           PGR10500
      CALL JGAP(0.)                                                     PGR10510
      CALL JJUST(1,1)                                                   PGR10520
      CALL JMOVE(0.,0.05*YWIND)                                         PGR10530
      IF(IRC.EQ.2) CALL J3STRG('CROSS SECTION ON ROW')                  PGR10540
      IF(IRC.EQ.1) CALL J3STRG('CROSS  SECTION ON COLUMN')              PGR10550
      WRITE (92,107) IHRC                                               PGR10560
      BACKSPACE 92                                                      PGR10570
      READ (92,113) AA2                                                 PGR10580
      X1=(22.0+(2-IRC)*2)*XLETR                                         PGR10590
      CALL JMOVE(X1,0.05*YWIND)                                         PGR10600
      CALL JJUST(1,1)                                                   PGR10610
      CALL J3STRG(AA2)                                                  PGR10620
C                                                                       PGR10630
  100 FORMAT(4I5,2F10.2)                                                PGR10640
  101 FORMAT(3F10.2)                                                    PGR10650
  102 FORMAT(F10.2)                                                     PGR10660
  108 FORMAT(F5.1)                                                      PGR10670
  103 FORMAT(A4)                                                        PGR10680
  107 FORMAT(I2)                                                        PGR10690
  109 FORMAT('#',I2)                                                    PGR10700
  110 FORMAT(F6.1)                                                      PGR10710
  113 FORMAT(A2)                                                        PGR10720
      CALL JCLOSE                                                       PGR10730
      CALL JPAUSE (1)                                                   PGR10740
      CALL JDEVOF (1)                                                   PGR10750
      CALL JDEND (1)                                                    PGR10760
      CALL JEND                                                         PGR10770
      RETURN                                                            PGR10780
      END                                                               PGR10790
```

4. GRAPHICS OUTPUT GENERATOR FOR THE VERTICAL CRATER RETREAT AFTER BLAST PROFILE

```
      PROGRAM PLBLVC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C   PURPOSE  :                                                         C
C      TO CREATE GRAPHICS OUTPUT FOR VCRM.                             C
C      GRAPHICS LANGUAGE : D13000                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CHARACTER*4 XALPHA
      CHARACTER*2 XALP
      CHARACTER*32 STUPID
      INTEGER HUES,SATS,LIGHTS,INDEXS,IWR
      DIMENSION HUES(24),SATS(24),LIGHTS(24),INDEXS(24),IWR(100)
      WRITE(7,3)
    3 FORMAT('  ENTER NUMBER OF HOLES IN THE BLAST :')
      READ(7,*) NUMBH
      WRITE(7,4)
    4 FORMAT('  ENTER MAX AND MIN FOR X AND Y COORDINATES OF THE STOPE '
     X,/ '  IN THE ORDER XMAX,XMIN,YMAX,YMIN :')
      READ (7,*) XMAX,XMIN,YMAX,YMIN
      WRITE(7,8)
    8 FORMAT('  ENTER ELEVATION OF THE FREE FACE :')
      READ(7,*) PLELEV
C-----------------------------------------------------------------------
C   **    CREATE SHADES OF THE SAME COLOR      **
C-----------------------------------------------------------------------
      INDEXS(1) = 10
      INDEXS(2) = 11
      INDEXS(3) = 12
      INDEXS(4) = 13
      INDEXS(5) = 14
      INDEXS(6) = 15
      INDEXS(7) = 16
      HUES(1) = 0
      HUES(2) = 0
      HUES(3) = 0
      HUES(4) = 0
      HUES(5) = 0
      HUES(6) = 0
      HUES(7) = 120
      SATS(1) = 19000
      SATS(2) = 19000
      SATS(3) = 19000
      SATS(4) = 19000
      SATS(5) = 19000
      SATS(6) = 19000
      SATS(7) = 30000
      LIGHTS(1)=32000
      LIGHTS(2)=28000
      LIGHTS(3)=27000
      LIGHTS(4)=26000
      LIGHTS(5)=25000
      LIGHTS(6)=23000
      LIGHTS(7)=20000
      DX = 0.7
```

```
      BY = 0.7
      CALL JBEGIN
      CALL JDINIT(1)
      CALL JDEVON(1)
      XM = XMIN - 10.0
      XMM = XMAX + 10.0
      YM = YMIN - 10.0
      YMM = YMAX + 10.0
      CALL JWINDO(XM,XMM,YM,YMM)
      CALL JCUTBL (1,7,INDEXS,HUES,SATS,LIGHTS)
      CALL JOPEN
      CALL JCOLOR(10)
      CALL JPINTR(1)
      CALL JPIDEX(10,9)
      CALL JRECT(XM,YM,XMM,YMM)
      CALL JMOVE(XMIN,YMIN)
      DO 110 I = 1,99
      KK = 1
C-----------------------------------------------------------------------
C   FILL QUADRILATERAL BLOCKS WITH A COLOR ACCORDING TO THE NUMBER
C       REPRESENTING THE DEPTH OF THE BLOCK.
C-----------------------------------------------------------------------
      READ(77,7) (IWR(JJ),JJ=1,100)
    7 FORMAT(80I1)
      DO 100 J = 1,99
      IF(IWR(J).EQ. 0 ) GO TO 10
      IF(IWR(J).EQ. 1 ) GO TO 20
      IF(IWR(J).EQ. 2 ) GO TO 30
      IF(IWR(J).EQ. 3 ) GO TO 40
      IF(IWR(J).EQ. 4 ) GO TO 50
      IF(IWR(J).EQ. 5 ) GO TO 60
      GO TO 100
   10 CALL JCOLOR(10)
      CALL JPINTR(1)
      CALL JPIDEX(10,20)
      GO TO 99
   20 CALL JCOLOR(11)
      CALL JPINTR(1)
      CALL JPIDEX(11,20)
      GO TO 99
   30 CALL JCOLOR(12)
      CALL JPINTR(1)
      CALL JPIDEX(12,20)
      GO TO 99
   40 CALL JCOLOR(13)
      CALL JPINTR(1)
      CALL JPIDEX(13,20)
      GO TO 99
   50 CALL JCOLOR(14)
      CALL JPINTR(1)
      CALL JPIDEX(14,20)
      GO TO 99
   60 CALL JCOLOR(15)
      CALL JPINTR(1)
      CALL JPIDEX(15,20)
```

```
   99 IF(IWK(J).NE.IWK(J+1)) GO TO 200                                  PLB01110
      GO TO 150                                                         PLB01120
  200 XX1 = XMIN + (FLOAT(KK-1))*DX                                     PLB01130
      YY1 = YMIN + (FLOAT(I-1)) *DY                                     PLB01140
      XX2 = XMIN + DX*FLOAT(J)                                          PLB01150
      YY2 = YMIN + DY*FLOAT(I)                                          PLB01160
      CALL JRECT(XX1,YY1,XX2,YY2)                                       PLB01170
      KK = J+1                                                          PLB01180
      GO TO 100                                                         PLB01190
  150 XX1 = XMIN + (FLOAT(KK-1))*DX                                     PLB01200
      YY1 = YMIN + (FLOAT(I-1))*DY                                      PLB01210
      XX2 = XMIN + DX*FLOAT(J)                                          PLB01220
      YY2 = YMIN + DY*FLOAT(I)                                          PLB01230
      IF(J.LT.99) GO TO 100                                             PLB01240
      CALL JRECT(XX1,YY1,XX2,YY2)                                       PLB01250
  100 CONTINUE                                                          PLB01260
  110 CONTINUE                                                          PLB01270
      CALL JCOLOR(16)                                                   PLB01280
C-----------------------------------------------------------------------PLB01290
C         DRAW AXES AND BOREHOLES                                       PLB01300
C-----------------------------------------------------------------------PLB01310
      CALL JMOVE(XMIN,YMIN)                                             PLB01320
      CALL JDRAW(XMAX,YMIN)                                             PLB01330
      CALL JMOVE(XMIN,YMIN)                                             PLB01340
      CALL JDRAW(XMIN,YMAX)                                             PLB01350
      DO 400 I = 1,10                                                   PLB01360
      XL = XMIN + DX * FLOAT(I-1) * 10.0                                PLB01370
      CALL JMOVE(XL,YMIN)                                               PLB01380
      CALL JDRAW(XL,YMIN-0.5)                                           PLB01390
      CALL JMOVE(XL,YMIN-1.0)                                           PLB01400
      IF(XL.GE.0) GO TO 112                                             PLB01410
      NL = IFIX(XL - 0.5)                                               PLB01420
      GO TO 113                                                         PLB01430
  112 NL = IFIX(XL + 0.5)                                               PLB01440
  113 WRITE(61,300) NL                                                  PLB01450
      BACKSPACE 61                                                      PLB01460
      READ(61,301) XALPHA                                               PLB01470
  301 FORMAT(A4)                                                        PLB01480
      CALL JJUST(2,3)                                                   PLB01490
      CALL JSIZE(1.0,1.0)                                               PLB01500
      CALL JSTRG(XALPHA)                                                PLB01510
  400 CONTINUE                                                          PLB01520
      DO 500 I = 1,10                                                   PLB01530
      YL = YMIN + DY*FLOAT(I-1)*10.0                                    PLB01540
      CALL JMOVE(XMIN,YL)                                               PLB01550
      CALL JDRAW(XMIN-0.5,YL)                                           PLB01560
      CALL JMOVE(XMIN-1.0,YL)                                           PLB01570
      IF(YL.GE.0.0) GO TO 412                                           PLB01580
      NL = IFIX(YL - 0.5)                                               PLB01590
      GO TO 413                                                         PLB01600
  412 NL = IFIX(YL + 0.5)                                               PLB01610
  413 WRITE(61,300) NL                                                  PLB01620
  300 FORMAT(I4)                                                        PLB01630
      BACKSPACE 61                                                      PLB01640
      READ(61,301) XALPHA                                               PLB01650
```

```
      CALL JJUST(3,2)                                                   PLB01660
      CALL JSIZE(1.0,1.0)                                               PLB01670
      CALL J3STRG(XALPHA)                                               PLB01680
  500 CONTINUE                                                          PLB01690
      CALL JCMARK(3)                                                    PLB01700
      DO 600 I = 1,NUMBH                                                PLB01710
      WRITE(61,510) I                                                   PLB01720
  510 FORMAT(I2)                                                        PLB01730
      BACKSPACE 61                                                      PLB01740
      READ(61,520) XALP                                                 PLB01750
  520 FORMAT(1A2)                                                       PLB01760
      READ(62,670) XX,YY,ZZ                                             PLB01770
  670 FORMAT(3F15.1)                                                    PLB01780
      CALL JMARK(XX,YY)                                                 PLB01790
      CALL JMOVE(XX+0.5,YY)                                             PLB01800
      CALL JJUST(1,2)                                                   PLB01810
      CALL JSIZE(1.0,1.0)                                               PLB01820
      CALL J3STRG(XALP)                                                 PLB01830
  600 CONTINUE                                                          PLB01840
      READ(51,610) STOPID                                               PLB01850
  610 FORMAT(A32)                                                       PLB01860
      CALL JMOVE((XM+XMM)/2.0,YMM-3.0)                                  PLB01870
      CALL JJUST(2,2)                                                   PLB01880
      CALL JSIZE(2.0,2.0)                                               PLB01890
      CALL J3STRG(STOPID)                                               PLB01900
      IELEV = IFIX(PLELEV)                                              PLB01910
      WRITE(61,630) IELEV                                               PLB01920
  630 FORMAT(I4)                                                        PLB01930
      BACKSPACE 61                                                      PLB01940
      READ(61,640) XALPHA                                               PLB01950
  640 FORMAT(A4)                                                        PLB01960
      CALL JMOVE((XM+XMM-12.)/2.0,YMM-6.0)                              PLB01970
      CALL JJUST(2,2)                                                   PLB01980
      CALL JSIZE(2.0,2.0)                                               PLB01990
      CALL J3STRG('PLAN ON ELEVATION ')                                 PLB02000
      CALL JMOVE((XM+XMM+36.0)/2.0,YMM-6.0)                             PLB02010
      CALL JJUST(2,2)                                                   PLB02020
      CALL JSIZE(2.0,2.0)                                               PLB02030
      CALL J3STRG(XALPHA)                                               PLB02040
      CALL JMOVE((XM+XMM)/2.0,YM+3.0)                                   PLB02050
      CALL JJUST(2,2)                                                   PLB02060
      CALL JSIZE(2.0,2.0)                                               PLB02070
      CALL J3STRG('X AXIS (FT)')                                        PLB02080
      CALL JBASE(0.0,1.0,0.0)                                           PLB02090
      CALL JMOVE(XM+3.0,(YM+YMM)/2.0)                                   PLB02100
      CALL JJUST(2,2)                                                   PLB02110
      CALL JSIZE(2.0,2.0)                                               PLB02120
      CALL J3STRG('Y AXIS (FT)')                                        PLB02130
      CALL JCLOSE                                                       PLB02140
      CALL JPAUSE(1)                                                    PLB02150
      CALL JDEVUE(1)                                                    PLB02160
      CALL JDEND(1)                                                     PLB02170
      CALL JEND                                                         PLB02180
      STOP                                                              PLB02190
      END                                                               PLB02200
```

APPENDIX B

PROGRAM INPUT DATA

1. LARGE DIAMETER BLASTHOLE BENCHING METHOD
2. VERTICAL CRATER RETREAT METHOD
3. EXEC FILES FOR THE TWO GRAPHICS PROGRAMS

1. LARGE DIAMETER BLASTHOLE BENCHING METHOD

```
EXEC DI3000 TEK4107
EXEC LINEMODE ON
FILEDEF 07 TERMINAL
FILEDEF 08 DISK BCURV DATA B
FILEDEF 09 DISK BBLAST DATA B
FILEDEF 11 DISK VIBRAT DATA B
FILEDEF 12 DISK TOP DATA B
FILEDEF 51 DISK INCOSTOP DATA B
FILEDEF 61 DISK JUNK DATA B
FILEDEF 66 DISK INCO62 DATA B
FILEDEF 70 DISK CHPLOT DATA B
FILEDEF 72 DISK OUTLINE DATA B
FILEDEF 06 DISK FRESULT LISTING B
FILEDEF 62 DISK PCH38 DATA B
EXEC EXECUTE LLL FORT77
EXEC LINEMODE OFF
```

```
 4
 0.0          4.0
12.0          4.0
12.0         50.0
 0.0         50.0
```

```
8
0.6000    0.1400
0.7100    0.2000
0.8300    0.2200
1.0700    0.2700
1.5900    0.3400
0.8700    0.5000
0.5200    0.4900
0.3600    0.6100
8
2.0000    0.1400
2.0200    0.2000
2.0300    0.2200
2.1300    0.2700
2.5000    0.3400
2.1000    0.5000
1.6700    0.4900
1.5000    0.6100
8
0.6000    0.1400
0.7100    0.2000
0.8300    0.2200
1.0700    0.2700
1.5900    0.3400
0.8700    0.5000
0.5200    0.4900
0.3600    0.6100
8
2.0000    0.1400
2.0200    0.2000
2.0300    0.2200
2.1300    0.2700
2.5000    0.3400
2.1000    0.5000
1.6700    0.4900
1.5000    0.6100
```

1
40.0 4.0 40.0 60.0

1	9	1	1	106.00
11340.70		13167.20		9443.98
11340.70		13167.20		9443.98
11340.78		13166.54		9453.96
11340.87		13165.92		9463.93
11341.04		13165.31		9473.91
11341.23		13164.82		9483.89
11341.43		13164.28		9493.87
11341.69		13163.69		9503.85
11341.92		13163.16		9513.83
11342.04		13162.65		9523.00
2	9	1	2	106.00
11331.80		13172.10		9444.03
11331.80		13172.10		9444.03
11331.84		13171.88		9454.03
11331.95		13171.44		9464.02
11332.09		13171.10		9474.01
11332.26		13170.75		9484.00
11332.40		13170.48		9493.99
11332.59		13170.12		9503.98
11332.79		13169.79		9513.97
11332.99		13169.50		9523.00
3	9	1	3	106.00
11323.00		13176.70		9444.00
11323.00		13176.70		9444.00
11323.25		13176.16		9453.98
11323.53		13175.76		9463.96
11323.82		13175.34		9473.95
11324.05		13174.93		9483.94
11324.28		13174.48		9493.92
11324.47		13173.94		9503.90
11324.71		13173.41		9513.88
11324.95		13172.93		9523.00
4	9	1	4	106.00
11315.00		13180.90		9444.63
11315.00		13180.90		9444.63
11315.00		13180.79		9454.62
11315.09		13180.49		9464.62
11315.15		13180.18		9474.61
11315.21		13179.87		9484.60
11315.28		13179.58		9494.59
11315.34		13179.27		9504.59
11315.48		13178.91		9514.58
11315.45		13178.65		9523.00
5	9	2	1	106.00
11344.60		13176.00		9444.63
11344.60		13176.00		9444.63
11344.76		13175.48		9454.61
11344.82		13175.10		9464.61
11344.90		13174.65		9474.59
11344.99		13174.21		9484.58
11345.08		13173.76		9494.57
11345.16		13173.37		9504.56
11345.32		13172.94		9514.55
11345.37		13172.47		9523.00

6	9	2	2	106.00
11337.00		13180.00		9444.63
11337.00		13180.00		9444.63
11337.09		13179.55		9454.62
11337.39		13179.08		9464.60
11337.62		13178.64		9474.59
11337.81		13178.18		9484.57
11337.90		13177.74		9494.56
11338.00		13177.31		9504.55
11338.09		13176.87		9514.54
11338.09		13176.45		9523.00
7	9	2	3	106.00
11327.90		13185.30		9444.63
11327.90		13185.30		9444.63
11328.09		13184.95		9454.62
11328.21		13184.56		9464.61
11328.26		13184.20		9474.60
11328.34		13183.82		9484.59
11328.42		13183.54		9494.58
11328.49		13183.19		9504.57
11328.59		13182.82		9514.57
11328.60		13182.49		9523.00
8	9	2	4	106.00
11319.60		13189.60		9444.63
11319.60		13189.60		9444.63
11319.65		13189.38		9454.62
11319.76		13189.05		9464.62
11319.81		13188.70		9474.61
11319.79		13188.42		9484.61
11319.70		13188.13		9494.60
11319.58		13187.82		9504.59
11319.54		13187.52		9514.58
11319.44		13187.26		9523.00
9	9	3	1	106.00
11349.40		13184.90		9444.63
11349.40		13184.90		9444.63
11349.61		13184.43		9454.61
11349.73		13184.09		9464.61
11349.84		13183.66		9474.59
11349.77		13183.27		9484.58
11349.67		13182.91		9494.57
11349.56		13182.56		9504.57
11349.43		13182.03		9514.55
11349.29		13181.56		9523.00
10	9	3	2	106.00
11341.00		13189.20		9444.63
11341.00		13189.20		9444.63
11341.12		13188.78		9454.62
11341.04		13188.31		9464.61
11340.96		13187.92		9474.59
11340.86		13187.55		9484.59
11340.71		13187.18		9494.57
11340.54		13186.77		9504.56
11340.42		13186.39		9514.55
11340.28		13186.03		9523.00

```
 11     9     3     3     106.00
11333.00  13193.90  9444.63
11333.00  13193.90  9444.63
11333.07  13193.52  9454.62
11332.94  13193.20  9464.61
11332.78  13192.79  9474.60
11332.54  13192.42  9484.59
11332.25  13192.07  9494.58
11332.01  13191.79  9504.57
11331.82  13191.53  9514.56
11331.64  13191.29  9523.00
 12     9     3     4     106.00
11326.00  13197.00  9444.63
11326.00  13197.00  9444.63
11326.07  13196.62  9454.62
11325.94  13196.30  9464.61
11325.78  13195.89  9474.60
11325.54  13195.52  9484.59
11325.25  13195.17  9494.58
11325.01  13194.89  9504.57
11324.82  13194.63  9514.56
11324.64  13194.39  9523.00
 99
```

2. VERTICAL CRATER RETREAT METHOD

```
&TRACE OFF
CLRSCRN
FILEDEF 07 TERM
FILEDEF 51 DISK INCOSTOP DATA B
FILEDEF 61 DISK INCO61 DATA B
FILEDEF 62 DISK PCH3 DATA B
FILEDEF 08 DISK CRCURV DATA B
FILEDEF 10 DISK SEQB DATA B
FILEDEF 11 DISK VIBRAT DATA B
FILEDEF 06 DISK FRESULV LISTING B
FILEDEF 70 DISK CHLOC DATA B
FILEDEF 77 DISK PLBLD DATA B
EXEC EXECUTE VCR FORT77 (INCO
```

```
SAMPLE STOPE FROM INCO
  11343.00  13140.00   9444.00          REF POINT
     70.00     50.00    200.00          ST LEN WID DEP
     62.50                              ANGLE XNEW TO EAST
   12
```

1 20	1 1	106.00
11340.70	13167.20	9443.98
11340.70	13167.20	9443.98
11340.78	13166.54	9453.96
11340.87	13165.92	9463.93
11341.04	13165.31	9473.91
11341.23	13164.82	9483.89
11341.43	13164.28	9493.87
11341.69	13163.69	9503.85
11341.92	13163.16	9513.83
11342.04	13162.65	9523.82
11342.16	13162.11	9533.80
11342.32	13161.55	9543.78
11342.49	13160.98	9553.76
11342.62	13160.46	9563.75
11342.75	13159.91	9573.73
11342.94	13159.35	9583.71
11343.09	13158.81	9593.70
11343.25	13158.35	9603.68
11343.38	13157.82	9613.66
11343.50	13157.51	9618.65
11343.55	13157.30	9624.64
2 18	1 2	106.00
11331.80	13172.10	9444.03
11331.80	13172.10	9444.03
11331.84	13171.88	9454.03
11331.95	13171.44	9464.02
11332.09	13171.10	9474.01
11332.26	13170.75	9484.00
11332.40	13170.48	9493.99
11332.59	13170.12	9503.98
11332.79	13169.79	9513.97
11332.99	13169.50	9523.96
11333.20	13169.18	9533.96
11333.48	13168.85	9543.95
11333.76	13168.59	9553.94
11334.04	13168.30	9563.93
11334.27	13167.97	9573.91
11334.54	13167.70	9583.91
11334.97	13167.43	9593.89
11335.38	13167.13	9603.87
11335.55	13166.96	9624.45
3 16	1 3	106.00
11323.00	13176.70	9444.00
11323.00	13176.70	9444.00
11323.25	13176.16	9453.98
11323.53	13175.76	9463.96
11323.82	13175.34	9473.95
11324.05	13174.93	9483.94
11324.28	13174.48	9493.92
11324.47	13173.94	9503.90
11324.71	13173.41	9513.88
11324.95	13172.93	9523.87
11325.17	13172.43	9533.85
11325.36	13172.00	9543.84

11325.60	13171.54	9553.82
11325.82	13170.95	9563.80
11326.11	13170.34	9573.78
11326.37	13169.75	9583.75
11327.47	13167.58	9624.45
4 19	1 4	106.00
11315.00	13180.90	9444.63
11315.00	13180.90	9444.63
11315.00	13180.79	9454.62
11315.09	13180.49	9464.62
11315.15	13180.18	9474.61
11315.21	13179.87	9484.60
11315.28	13179.58	9494.59
11315.34	13179.27	9504.59
11315.48	13178.91	9514.58
11315.45	13178.65	9524.57
11315.46	13178.34	9534.57
11315.49	13178.02	9544.56
11315.46	13177.73	9554.55
11315.41	13177.36	9564.54
11315.40	13176.98	9574.54
11315.43	13176.59	9584.53
11315.43	13176.24	9594.52
11315.38	13175.86	9604.51
11315.32	13175.51	9614.50
11315.33	13175.14	9624.50
5 19	2 1	106.00
11344.60	13176.00	9444.63
11344.60	13176.00	9444.63
11344.76	13175.48	9454.61
11344.82	13175.10	9464.61
11344.90	13174.65	9474.59
11344.99	13174.21	9484.58
11345.08	13173.76	9494.57
11345.16	13173.37	9504.56
11345.32	13172.94	9514.55
11345.37	13172.47	9524.54
11345.39	13172.00	9534.52
11345.43	13171.60	9544.51
11345.49	13171.14	9554.50
11345.50	13170.72	9564.49
11345.50	13170.18	9574.47
11345.60	13169.68	9584.46
11345.55	13169.23	9594.45
11345.53	13168.67	9604.43
11345.44	13168.30	9614.42
11345.40	13167.70	9624.40
6 19	2 2	106.00
11337.00	13180.00	9444.63
11337.00	13180.00	9444.63
11337.09	13179.55	9454.62
11337.39	13179.08	9464.60
11337.62	13178.64	9474.59
11337.81	13178.18	9484.57
11337.90	13177.74	9494.56

11338.00	13177.31	9504.55
11338.09	13176.87	9514.54
11338.09	13176.45	9524.52
11338.06	13176.09	9534.52
11338.03	13175.69	9544.50
11337.95	13175.24	9554.49
11337.87	13174.71	9564.48
11337.75	13174.09	9574.45
11337.64	13173.54	9584.44
11337.43	13173.00	9594.42
11337.21	13172.36	9604.39
11337.03	13171.75	9614.37
11336.79	13171.04	9624.34
7 19	2 3	106.00
11327.90	13185.30	9444.63
11327.90	13185.30	9444.63
11328.09	13184.95	9454.62
11328.21	13184.56	9464.61
11328.26	13184.20	9474.60
11328.34	13183.82	9484.59
11328.42	13183.54	9494.58
11328.49	13183.19	9504.57
11328.59	13182.82	9514.57
11328.60	13182.49	9524.56
11328.64	13182.20	9534.55
11328.75	13181.72	9544.54
11328.77	13181.42	9554.53
11328.75	13181.04	9564.52
11328.74	13180.56	9574.51
11328.75	13180.09	9584.50
11328.83	13179.61	9594.48
11328.84	13179.22	9604.47
11328.82	13178.78	9614.46
11328.83	13178.26	9624.45
8 19	2 4	106.00
11319.60	13189.60	9444.63
11319.60	13189.60	9444.63
11319.65	13189.38	9454.62
11319.76	13189.05	9464.62
11319.81	13188.70	9474.61
11319.79	13188.42	9484.61
11319.70	13188.13	9494.60
11319.58	13187.82	9504.59
11319.54	13187.52	9514.58
11319.44	13187.26	9524.58
11319.19	13186.86	9534.57
11318.95	13186.51	9544.55
11318.59	13186.10	9554.54
11318.24	13185.67	9564.52
11317.80	13185.22	9574.50
11317.40	13184.75	9584.48
11316.96	13184.33	9594.46
11316.45	13183.80	9604.43
11315.94	13183.32	9614.41
11315.39	13182.86	9624.38

9	19	3	1	106.00
11349.40	13184.90	9444.63
11349.40	13184.90	9444.63
11349.61	13184.43	9454.61
11349.73	13184.09	9464.61
11349.84	13183.66	9474.59
11349.77	13183.27	9484.58
11349.67	13182.91	9494.57
11349.56	13182.56	9504.57
11349.43	13182.03	9514.55
11349.29	13181.56	9524.54
11349.16	13181.18	9534.52
11349.00	13180.86	9544.52
11348.87	13180.43	9554.50
11348.72	13180.02	9564.49
11348.57	13179.70	9574.48
11348.41	13179.25	9584.47
11348.26	13178.89	9594.46
11348.11	13178.41	9604.45
11347.96	13178.05	9614.44
11347.83	13177.60	9624.43
10	19	3	2	106.00
11341.00	13189.20	9444.63
11341.00	13189.20	9444.63
11341.12	13188.78	9454.62
11341.04	13188.31	9464.61
11340.96	13187.92	9474.59
11340.86	13187.55	9484.59
11340.71	13187.18	9494.57
11340.54	13186.77	9504.56
11340.42	13186.39	9514.55
11340.28	13186.03	9524.54
11340.13	13185.65	9534.53
11340.03	13185.30	9544.52
11339.94	13184.90	9554.51
11339.79	13184.49	9564.50
11339.64	13184.15	9574.49
11339.60	13183.77	9584.48
11339.45	13183.45	9594.48
11339.31	13183.03	9604.46
11339.18	13182.69	9614.46
11339.12	13182.32	9624.45
11	19	3	3	106.00
11333.00	13193.90	9444.63
11333.00	13193.90	9444.63
11333.07	13193.52	9454.62
11332.94	13193.20	9464.61
11332.78	13192.79	9474.60
11332.54	13192.42	9484.59
11332.25	13192.07	9494.58
11332.01	13191.79	9504.57
11331.82	13191.53	9514.56
11331.64	13191.29	9524.55
11331.46	13191.08	9534.55
11331.23	13190.88	9544.54

11331.03	13190.59	9554.54
11330.80	13190.37	9564.53
11330.58	13190.20	9574.52
11330.38	13190.00	9584.52
11330.15	13189.80	9594.51
11329.98	13189.60	9604.51
11329.73	13189.47	9614.50
11329.41	13189.29	9624.50
12 19	3 4	106.00
11326.00	13197.00	9444.63
11326.00	13197.00	9444.63
11326.07	13196.62	9454.62
11325.94	13196.30	9464.61
11325.78	13195.89	9474.60
11325.54	13195.52	9484.59
11325.25	13195.17	9494.58
11325.01	13194.89	9504.57
11324.82	13194.63	9514.56
11324.64	13194.39	9524.55
11324.46	13194.18	9534.55
11324.23	13193.98	9544.54
11324.03	13193.70	9554.54
11323.80	13193.47	9564.53
11323.58	13193.30	9574.52
11323.38	13193.10	9584.52
11323.15	13192.91	9594.51
11322.98	13192.70	9604.51
11322.73	13192.57	9614.50
11322.41	13192.39	9624.50
99		

```
8
0.6000     0.1400
0.7100     0.2000
0.8300     0.2200
1.0700     0.2700
1.5900     0.3400
0.8700     0.5000
0.5200     0.4900
0.3600     0.6100
8
2.0000     0.1400
2.0200     0.2000
2.0300     0.2200
2.1300     0.2700
2.5000     0.3400
2.1000     0.5000
1.6700     0.4900
1.5000     0.6100
```

3. EXEC FILES FOR THE TWO GRAPHICS PROGRAMS

```
FILEDEF 07 TERMINAL
FILEDEF 77 DISK PLBLD DATA B
FILEDEF 61 DISK TEMF1 DATA B
FILEDEF 62 DISK BLAST DATA B
FILEDEF 51 DISK INCOSTOP DATA B
GLOBAL TXTLIB
EXEC LINEMODE ON
EXEC D13000 TEK4107
LOAD PLBLAST ( START NOMAP CLEAR NOAUTO
EXEC LINEMODE OFF
```

```
EXEC DI3000 TEK4107
EXEC LINEMODE ON
FILEDEF 05 TERM
FILEDEF 06 TERM
FILEDEF 51 DISK INCUSTOP DATA B
FILEDEF 61 DISK INCD62 DATA B
FILEDEF 62 DISK PCH3 DATA B
FILEDEF 92 DISK P92 DATA B
EXEC EXECUTE PGRAPH  FORT77 ( INC77
EXEC LINEMODE OFF
*ERASE P92 DATA B
&EXIT
```

APPENDIX C

PROGRAM OUTPUT FOR

1. LARGE DIAMETER BLASTHOLE BENCHING METHOD

2. VERTICAL CRATER RETREAT METHOD

1. LARGE DIAMETER BLASTHOLE BENCHING METHOD

PROGRAM LDBB

WRITTEN FOR THE DESIGN OF THE BLAST IN AN UNDERGROUND STOPE USING THE LARGE DIAMETER BLASTHOLE BENCHING METHOD

SAMPLE STOPE

EXPLOSIVE USED :AQUAMEX

DENSITY :0.0540 LBS/CUB. INCH

DIAMETER :6.5000 INCHES

INITIATION SEQUENCE IN LEVEL # 1

HOLE NUMBER	DELAY PERIOD
1	1
2	2
3	3
4	4
5	5
6	4
7	5
8	6
9	7
10	6
11	7
12	8

WEIGHT OF EXPLOSIVE DECKS IN LEVEL # 1

HOLE NUMBER	WEIGHT (LBS)
1	122.
2	122.
3	122.
4	122.
5	122.
6	122.
7	146.
8	122.
9	122.
10	122.
11	122.
12	122.

PLAN VIEW ON ELEVATION 9520.0000

INITIATION SEQUENCE IN LEVEL # 2

HOLE NUMBER	DELAY PERIOD
1	3
2	4
3	5
4	6
5	7
6	6
7	7
8	8
9	9
10	8
11	9
12	10

WEIGHT OF EXPLOSIVE DECKS IN LEVEL # 2

HOLE NUMBER	WEIGHT (LBS)
1	122.
2	122.
3	122.
4	122.
5	122.
6	122.
7	146.
8	122.
9	122.
10	122.
11	122.
12	122.

PLAN VIEW ON ELEVATION   9504.7969

A146

```
 1.        21.       41.       61.       81.       0.        0.        0.
 +---------+---------+---------+---------+---------+---------+---------+--------
+
|
| ......
| .......
| ........
| .........A
| ...............A   A
| ....................
| ...................
| ...................
+ ...............A...
| ..........A........A
| ...................
| ...................
| ...................
| ..........A.........
| ...............A....A
| ....................
| ....................
| ....................
+ ...................A
| ..........A    A
| ........
| ......
| ......
|
|
|
|
+
|
|
|
|
|
|
|
|
|
+
|
|
|
|
|
|
|
|
|
```

INITIATION SEQUENCE IN LEVEL # 3

HOLE NUMBER	DELAY PERIOD
1	5
2	6
3	7
4	8
5	9
6	8
7	9
8	10
9	11
10	10
11	11
12	12

WEIGHT OF EXPLOSIVE DECKS IN LEVEL # 3

HOLE NUMBER	WEIGHT (LBS)
1	122.
2	122.
3	122.
4	122.
5	122.
6	122.
7	146.
8	122.
9	122.
10	122.
11	122.
12	122.

PLAN VIEW ON ELEVATION 9489.5937

INITIATION SEQUENCE IN LEVEL # 4

HOLE NUMBER	DELAY PERIOD
1	7
2	8
3	9
4	10
5	11
6	10
7	11
8	12
9	13
10	12
11	13
12	14

WEIGHT OF EXPLOSIVE DECKS IN LEVEL # 4

HOLE NUMBER	WEIGHT (LBS)
1	122.
2	122.
3	122.
4	122.
5	122.
6	122.
7	146.
8	122.
9	122.
10	122.
11	122.
12	122.

PLAN VIEW OR ELEVATION 9474.3906

INITIATION SEQUENCE IN LEVEL # 5

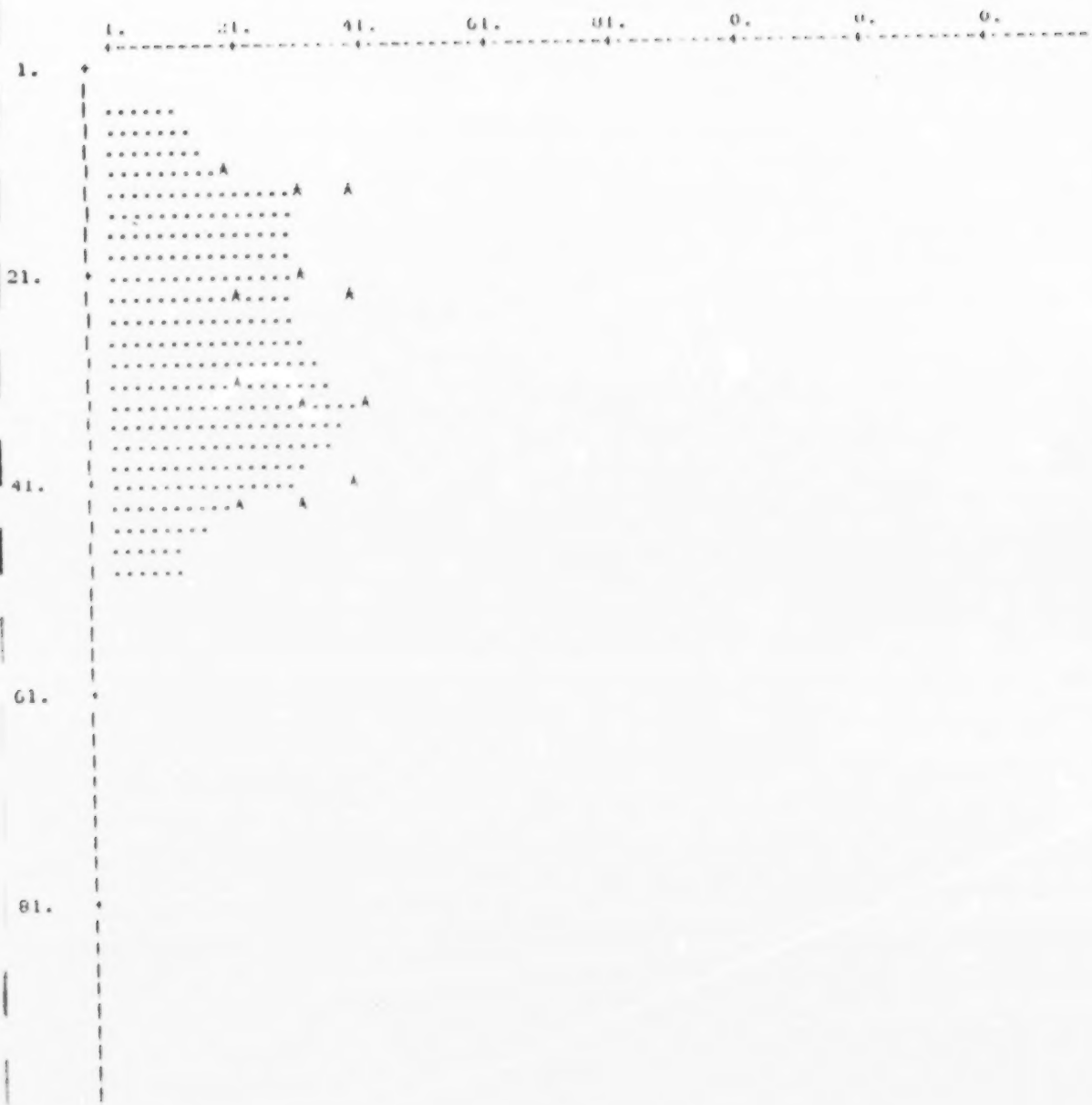
HOLE NUMBER	DELAY PERIOD
1	9
2	10
3	11
4	12
5	13
6	12
7	13
8	14
9	15
10	14
11	15
12	16

WEIGHT OF EXPLOSIVE DECKS IN LEVEL # 5

HOLE NUMBER	WEIGHT (LBS)
1	122.
2	122.
3	122.
4	122.
5	122.
6	146.
7	126.
8	105.
9	0.
10	0.
11	146.
12	0.

DATION 9459 1875

PLAN VIEW ON ELEVATION 9459.1875



SUMMARY

TOTAL WEIGHT OF EXPLOSIVE : 7066. LBS

DELAY NUMBER	QUANTITY
1	1
2	1
3	2
4	3
5	4
6	5
7	6
8	6
9	6
10	6
11	5
12	5
13	4
14	3
15	2
16	1

INSTRUCTIONS FOR LOADING

HOLE NUMBER	CHARGE LOCATION	CHARGE WEIGHT	DELAY NUMBER
1	79.	122.	1
	64.	122.	3
	49.	122.	5
	33.	122.	7
	18.	122.	9
2	79.	122.	2
	64.	122.	4
	48.	122.	6
	33.	122.	8
	18.	122.	10
3	79.	122.	3
	64.	122.	5
	48.	122.	7
	33.	122.	9
	18.	122.	11
4	78.	122.	4
	63.	122.	6
	48.	122.	8
	33.	122.	10
	17.	122.	12
5	78.	122.	5
	63.	122.	7
	48.	122.	9
	33.	122.	11
	17.	122.	13
6	78.	122.	4
	63.	122.	6
	48.	122.	8
	33.	122.	10
	18.	146.	12
7	79.	146.	5
	64.	146.	7
	48.	146.	9
	33.	146.	11
	18.	126.	13
8	78.	122.	6
	63.	122.	8
	48.	122.	10

	17.	122.	13
6	78.	122.	4
	63.	122.	6
	48.	122.	8
	33.	122.	10
	18.	146.	12
7	79.	146.	5
	64.	146.	7
	48.	146.	9
	33.	146.	11
	18.	126.	13
8	78.	122.	6
	63.	122.	8
	48.	122.	10
	33.	122.	12
	17.	105.	14
9	78.	122.	7
	63.	122.	9
	48.	122.	11
	33.	122.	13
	15.	0.	15
10	78.	122.	6
	63.	122.	8
	48.	122.	10
	33.	122.	12
	15.	0.	14
11	78.	122.	7
	63.	122.	9
	48.	122.	11
	33.	122.	13
	18.	146.	15
12	78.	122.	8
	63.	122.	10
	48.	122.	12
	33.	122.	14
	15.	0.	16

2. VERTICAL CRATER RETREAT METHOD

PROGRAM VCRM

WRITTEN FOR THE DESIGN OF THE BLAST IN AN UNDERGROUND STOPE USING THE VERTICAL CRATER RETREAT METHOD.

TITLE

DATE : 11/02/84

TIME : 16:16:25

EXPLOSIVE USED :Q

DENSITY :0.0540 LBS/CUB. INCH

DIAMETER :6.5000 INCHES

MODIFIED INITIATION SEQUENCE

HOLE NUMBER	DELAY PERIOD
1	1
2	2
3	3
4	5
5	2
6	3
7	5
8	6
9	2
10	4
11	4
12	6

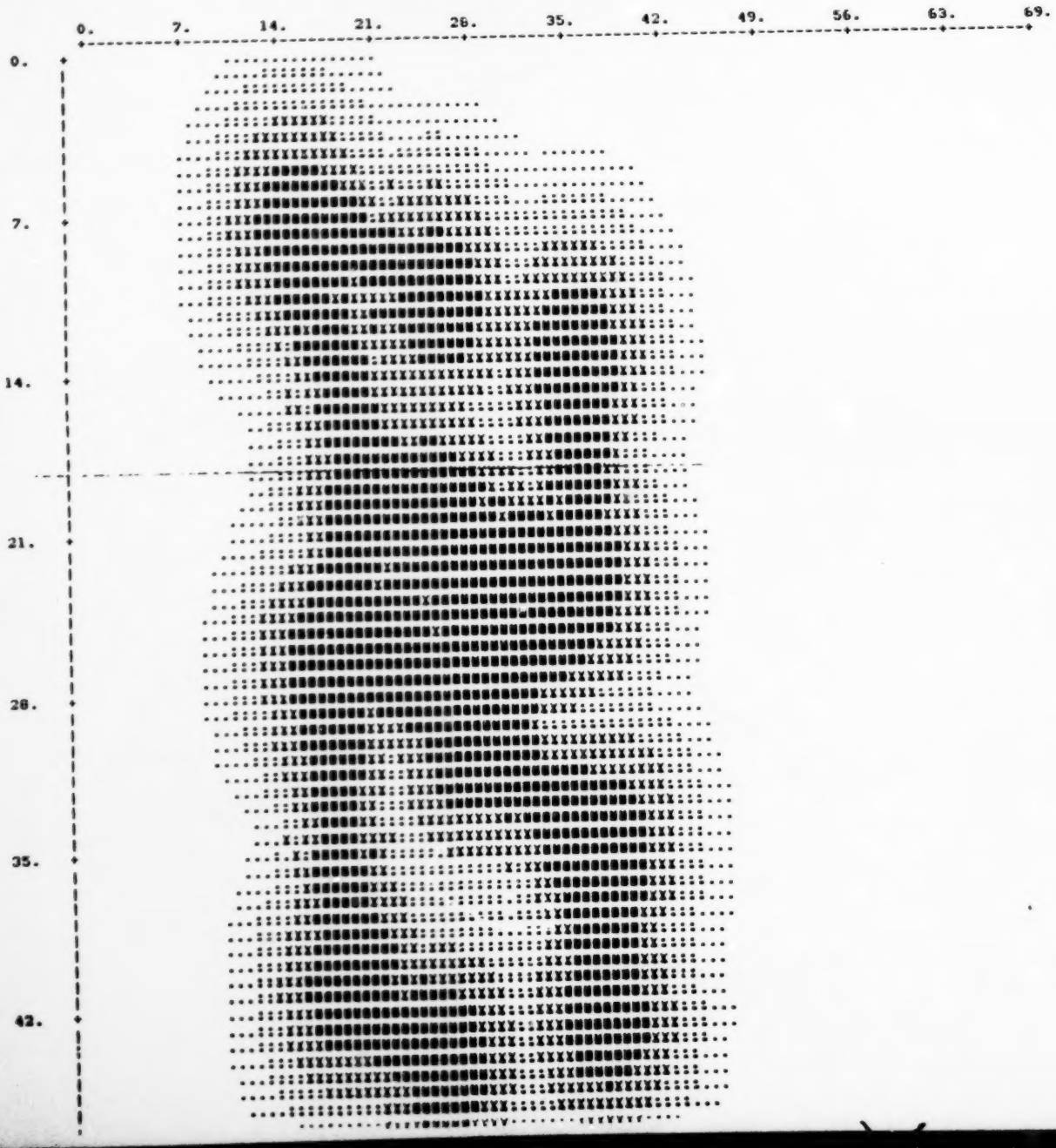
WEIGHT OF CHARGES IN THE BLAST

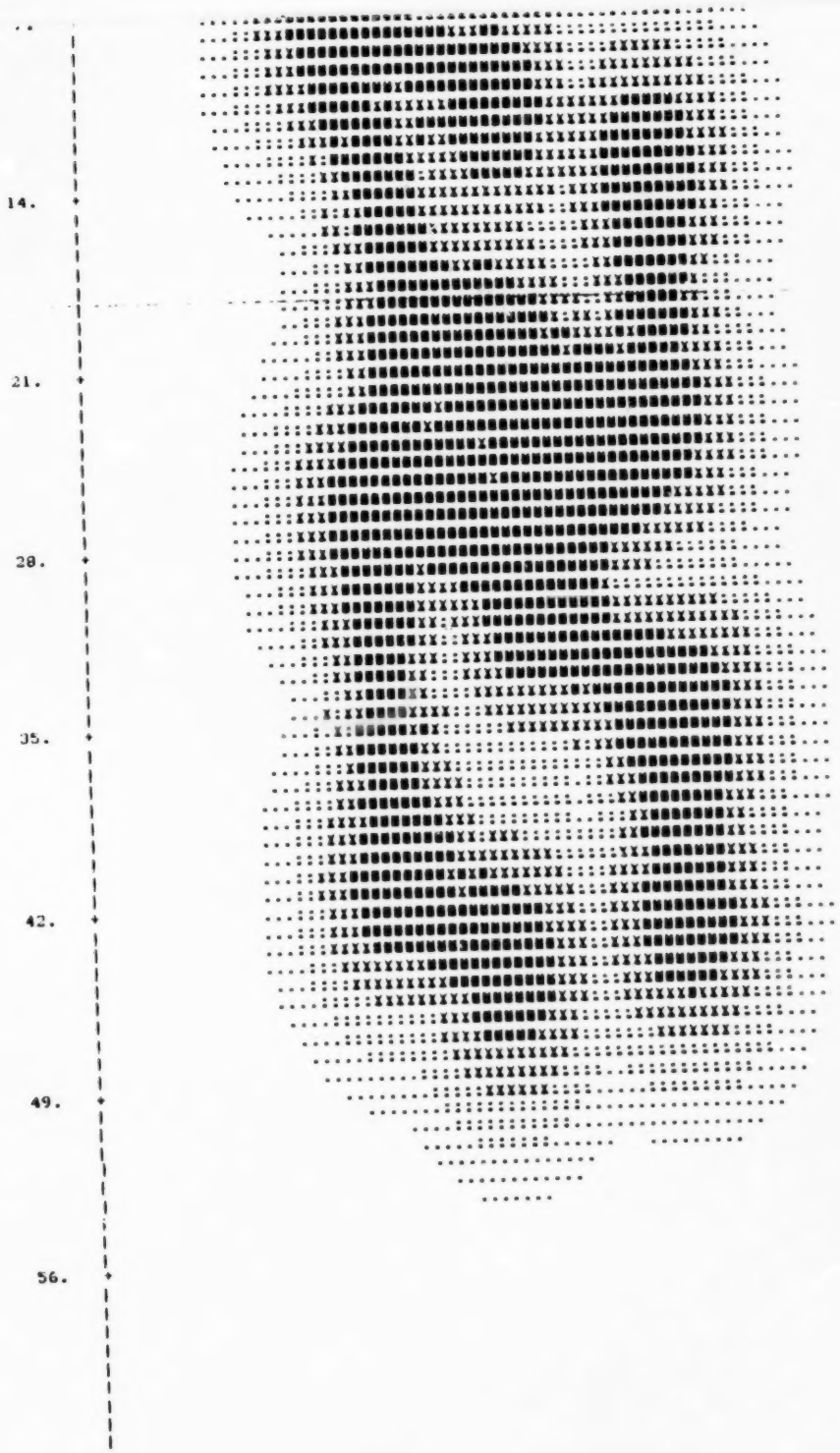
HOLE NUMBER	WEIGHT (LBS)
1	70.
2	70.
3	70.
4	70.
5	70.
6	70.
7	70.
8	70.
9	70.
10	70.
11	70.
12	70.

PLAN VIEW OF THE BLAST

ELEVATIONS REPRESENTED BY THE PRINTOUT

	180.0 FT
.....	178.4 FT
::::::::	176.9 FT
XXXXXXXX	175.3 FT
********	173.8 FT
********	172.2 FT

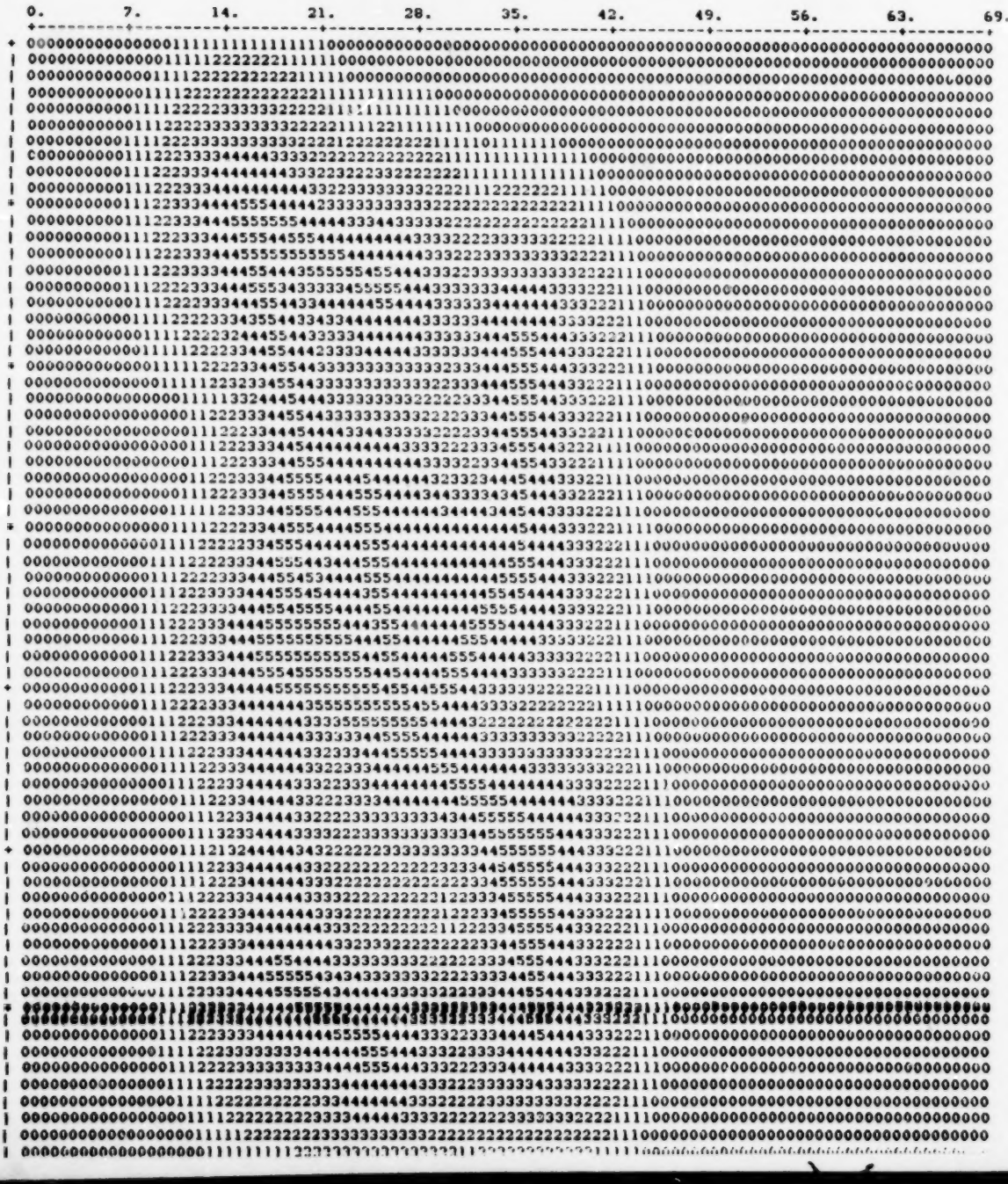


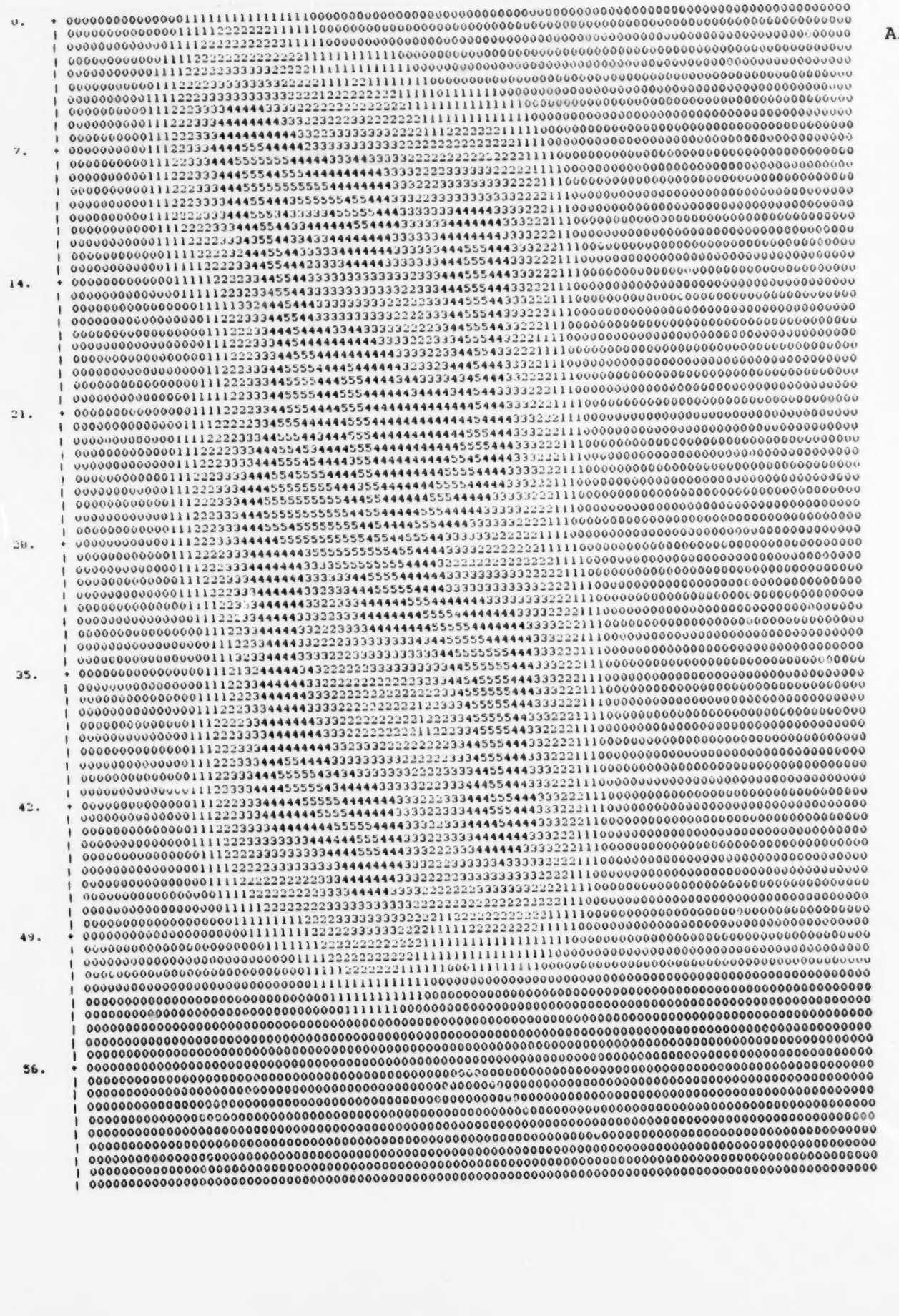

14.
21.
28.
35.
42.
49.
56.

PLAN VIEW OF THE BLAST

ELEVATIONS REPRESENTED BY THE NUMBERS OF THE PRINTOUT

NUMBER 0	180.0	FT
NUMBER 1	178.4	FT
NUMBER 2	176.9	FT
NUMBER 3	175.3	FT
NUMBER 4	173.8	FT
NUMBER 5	172.2	FT





SUMMARY

TOTAL WEIGHT OF EXPLOSIVE : 839.

DELAY NUMBER	QUANTITY
1	1
2	3
3	2
4	2
5	2
6	2

INSTRUCTIONS FOR LOADING

HOLE NUMBER	CHARGE LOCATION	CHARGE WEIGHT	DELAY NUMBER
1	174.	70.	1
2	174.	70.	2
3	174.	70.	3
4	173.	70.	5
5	173.	70.	2
6	173.	70.	3
7	173.	70.	5
8	173.	70.	6
9	173.	70.	2
10	173.	70.	4
11	173.	70.	4
12	173.	70.	6

POOR

QUALITY

ORIGINAL